

# NVIDIA Performance Primitives (NPP)

Version 4.0

March 14, 2014



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>NVIDIA Performance Primitives</b>               | <b>1</b>  |
| 1.1      | What is NPP?                                       | 1         |
| 1.2      | Documentation                                      | 1         |
| 1.3      | Technical Specifications                           | 2         |
| 1.4      | Files  | 2         |
| 1.4.1    | Header Files                                       | 2         |
| 1.4.2    | Library Files                                      | 2         |
| 1.5      | Supported NVIDIA Hardware                          | 3         |
| <b>2</b> | <b>General API Conventions</b>                     | <b>5</b>  |
| 2.1      | Memory Management                                  | 6         |
| 2.1.1    | Scratch Buffer and Host Pointer                    | 6         |
| 2.2      | Function Naming                                    | 7         |
| 2.3      | Integer Result Scaling                             | 7         |
| 2.4      | Rounding Modes                                     | 8         |
| 2.4.1    | Rounding Mode Parameter                            | 8         |
| <b>3</b> | <b>Signal-Processing Specific API Conventions</b>  | <b>9</b>  |
| 3.1      | Signal Data  | 10        |
| 3.1.1    | Parameter Names for Signal Data                    | 10        |
| 3.1.1.1  | Source Signal Pointer                              | 10        |
| 3.1.1.2  | Destination Signal Pointer                         | 10        |
| 3.1.1.3  | In-Place Signal Pointer                            | 10        |
| 3.1.2    | Signal Data Alignment Requirements                 | 11        |
| 3.1.3    | Signal Data Related Error Codes                    | 11        |
| 3.2      | Signal Length                                      | 11        |
| 3.2.1    | Length Related Error Codes                         | 11        |
| <b>4</b> | <b>Imaging-Processing Specific API Conventions</b> | <b>13</b> |

|          |  |           |
|----------|--|-----------|
| 4.1      | Function Naming                          | 14        |
| 4.2      | Image Data                               | 14        |
| 4.2.1    | Line Step                                | 15        |
| 4.2.2    | Parameter Names for Image Data           | 15        |
| 4.2.2.1  | Passing Source-Image Data                | 15        |
| 4.2.2.2  | Passing Destination-Image Data           | 16        |
| 4.2.2.3  | Passing In-Place Image Data              | 18        |
| 4.2.2.4  | Passing Mask-Image Data                  | 18        |
| 4.2.2.5  | Passing Channel-of-Interest Data         | 18        |
| 4.2.3    | Image Data Alignment Requirements        | 18        |
| 4.2.4    | Image Data Related Error Codes           | 19        |
| 4.3      | Region-of-Interest (ROI)                 | 19        |
| 4.3.1    | ROI Related Error Codes                  | 19        |
| 4.4      | Masked Operation                         | 20        |
| 4.5      | Channel-of-Interest API                  | 20        |
| 4.5.1    | Select-Channel Source-Image Pointer      | 20        |
| 4.5.2    | Select-Channel Source-Image              | 20        |
| 4.5.3    | Select-Channel Destination-Image Pointer | 20        |
| 4.6      | Source-Image Sampling                    | 21        |
| 4.6.1    | Point-Wise Operations                    | 21        |
| 4.6.2    | Neighborhood Operations                  | 21        |
| 4.6.2.1  | Mask-Size Parameter                      | 21        |
| 4.6.2.2  | Anchor-Point Parameter                   | 22        |
| 4.6.2.3  | Sampling Beyond Image Boundaries         | 22        |
| <b>5</b> | <b>Module Index</b>                      | <b>23</b> |
| 5.1      | Modules                                  | 23        |
| <b>6</b> | <b>Data Structure Index</b>              | <b>29</b> |
| 6.1      | Data Structures                          | 29        |
| <b>7</b> | <b>Module Documentation</b>              | <b>31</b> |
| 7.1      | NPP Core                                 | 31        |
| 7.1.1    | Detailed Description                     | 31        |
| 7.1.2    | Function Documentation                   | 32        |
| 7.1.2.1  | nppGetGpuComputeCapability               | 32        |
| 7.1.2.2  | nppGetGpuName                            | 32        |
| 7.1.2.3  | nppGetGpuNumSMs                          | 32        |



|          |  |    |
|----------|--|----|
| 7.1.2.4  | <a href="#">nppGetLibVersion</a>                   | 32 |
| 7.1.2.5  | <a href="#">nppGetMaxThreadsPerBlock</a>           | 32 |
| 7.1.2.6  | <a href="#">nppGetMaxThreadsPerSM</a>              | 33 |
| 7.1.2.7  | <a href="#">nppGetStream</a>                       | 33 |
| 7.1.2.8  | <a href="#">nppSetStream</a>                       | 33 |
| 7.2      | <a href="#">NPP Type Definitions and Constants</a> | 34 |
| 7.2.1    | <a href="#">Define Documentation</a>               | 39 |
| 7.2.1.1  | <a href="#">NPP_MAX_16S</a>                        | 39 |
| 7.2.1.2  | <a href="#">NPP_MAX_16U</a>                        | 39 |
| 7.2.1.3  | <a href="#">NPP_MAX_32S</a>                        | 39 |
| 7.2.1.4  | <a href="#">NPP_MAX_32U</a>                        | 39 |
| 7.2.1.5  | <a href="#">NPP_MAX_64S</a>                        | 39 |
| 7.2.1.6  | <a href="#">NPP_MAX_64U</a>                        | 39 |
| 7.2.1.7  | <a href="#">NPP_MAX_8S</a>                         | 40 |
| 7.2.1.8  | <a href="#">NPP_MAX_8U</a>                         | 40 |
| 7.2.1.9  | <a href="#">NPP_MAXABS_32F</a>                     | 40 |
| 7.2.1.10 | <a href="#">NPP_MAXABS_64F</a>                     | 40 |
| 7.2.1.11 | <a href="#">NPP_MIN_16S</a>                        | 40 |
| 7.2.1.12 | <a href="#">NPP_MIN_16U</a>                        | 40 |
| 7.2.1.13 | <a href="#">NPP_MIN_32S</a>                        | 40 |
| 7.2.1.14 | <a href="#">NPP_MIN_32U</a>                        | 40 |
| 7.2.1.15 | <a href="#">NPP_MIN_64S</a>                        | 40 |
| 7.2.1.16 | <a href="#">NPP_MIN_64U</a>                        | 40 |
| 7.2.1.17 | <a href="#">NPP_MIN_8S</a>                         | 40 |
| 7.2.1.18 | <a href="#">NPP_MIN_8U</a>                         | 41 |
| 7.2.1.19 | <a href="#">NPP_MINABS_32F</a>                     | 41 |
| 7.2.1.20 | <a href="#">NPP_MINABS_64F</a>                     | 41 |
| 7.2.2    | <a href="#">Enumeration Type Documentation</a>     | 41 |
| 7.2.2.1  | <a href="#">NppCmpOp</a>                           | 41 |
| 7.2.2.2  | <a href="#">NppGpuComputeCapability</a>            | 41 |
| 7.2.2.3  | <a href="#">NppHintAlgorithm</a>                   | 41 |
| 7.2.2.4  | <a href="#">NppiAlphaOp</a>                        | 42 |
| 7.2.2.5  | <a href="#">NppiAxis</a>                           | 42 |
| 7.2.2.6  | <a href="#">NppiBorderType</a>                     | 42 |
| 7.2.2.7  | <a href="#">NppiHuffmanTableType</a>               | 42 |
| 7.2.2.8  | <a href="#">NppiInterpolationMode</a>              | 43 |

|          |                                   |    |
|----------|-----------------------------------|----|
| 7.2.2.9  | NppiMaskSize                      | 43 |
| 7.2.2.10 | NppRoundMode                      | 43 |
| 7.2.2.11 | NppStatus                         | 44 |
| 7.2.2.12 | NppsZCType                        | 46 |
| 7.3      | Basic NPP Data Types              | 47 |
| 7.3.1    | Typedef Documentation             | 48 |
| 7.3.1.1  | Npp16s                            | 48 |
| 7.3.1.2  | Npp16u                            | 48 |
| 7.3.1.3  | Npp32f                            | 48 |
| 7.3.1.4  | Npp32fc                           | 48 |
| 7.3.1.5  | Npp32s                            | 48 |
| 7.3.1.6  | Npp32sc                           | 49 |
| 7.3.1.7  | Npp32u                            | 49 |
| 7.3.1.8  | Npp32uc                           | 49 |
| 7.3.1.9  | Npp64f                            | 49 |
| 7.3.1.10 | Npp64fc                           | 49 |
| 7.3.1.11 | Npp64s                            | 49 |
| 7.3.1.12 | Npp64sc                           | 49 |
| 7.3.1.13 | Npp64u                            | 49 |
| 7.3.1.14 | Npp8s                             | 49 |
| 7.3.1.15 | Npp8u                             | 49 |
| 7.3.2    | Function Documentation            | 49 |
| 7.3.2.1  | __align__                         | 49 |
| 7.3.2.2  | __align__                         | 50 |
| 7.3.3    | Variable Documentation            | 50 |
| 7.3.3.1  | Npp16sc                           | 50 |
| 7.3.3.2  | Npp16uc                           | 50 |
| 7.3.3.3  | Npp8uc                            | 50 |
| 7.4      | NPP Image Processing              | 51 |
| 7.5      | Arithmetic and Logical Operations | 52 |
| 7.6      | Arithmetic Operations             | 53 |
| 7.7      | AddC                              | 55 |
| 7.7.1    | Detailed Description              | 60 |
| 7.7.2    | Function Documentation            | 60 |
| 7.7.2.1  | nppiAddC_16s_AC4IRSfs             | 60 |
| 7.7.2.2  | nppiAddC_16s_AC4RSfs              | 60 |

|          |  |    |
|----------|--|----|
| 7.7.2.3  | <a href="#">nppiAddC_16s_C1IRSfs</a>   | 60 |
| 7.7.2.4  | <a href="#">nppiAddC_16s_C1RSfs</a>    | 61 |
| 7.7.2.5  | <a href="#">nppiAddC_16s_C3IRSfs</a>   | 61 |
| 7.7.2.6  | <a href="#">nppiAddC_16s_C3RSfs</a>    | 62 |
| 7.7.2.7  | <a href="#">nppiAddC_16s_C4IRSfs</a>   | 62 |
| 7.7.2.8  | <a href="#">nppiAddC_16s_C4RSfs</a>    | 62 |
| 7.7.2.9  | <a href="#">nppiAddC_16sc_AC4IRSfs</a> | 63 |
| 7.7.2.10 | <a href="#">nppiAddC_16sc_AC4RSfs</a>  | 63 |
| 7.7.2.11 | <a href="#">nppiAddC_16sc_C1IRSfs</a>  | 64 |
| 7.7.2.12 | <a href="#">nppiAddC_16sc_C1RSfs</a>   | 64 |
| 7.7.2.13 | <a href="#">nppiAddC_16sc_C3IRSfs</a>  | 64 |
| 7.7.2.14 | <a href="#">nppiAddC_16sc_C3RSfs</a>   | 65 |
| 7.7.2.15 | <a href="#">nppiAddC_16u_AC4IRSfs</a>  | 65 |
| 7.7.2.16 | <a href="#">nppiAddC_16u_AC4RSfs</a>   | 66 |
| 7.7.2.17 | <a href="#">nppiAddC_16u_C1IRSfs</a>   | 66 |
| 7.7.2.18 | <a href="#">nppiAddC_16u_C1RSfs</a>    | 66 |
| 7.7.2.19 | <a href="#">nppiAddC_16u_C3IRSfs</a>   | 67 |
| 7.7.2.20 | <a href="#">nppiAddC_16u_C3RSfs</a>    | 67 |
| 7.7.2.21 | <a href="#">nppiAddC_16u_C4IRSfs</a>   | 68 |
| 7.7.2.22 | <a href="#">nppiAddC_16u_C4RSfs</a>    | 68 |
| 7.7.2.23 | <a href="#">nppiAddC_32f_AC4IR</a>     | 68 |
| 7.7.2.24 | <a href="#">nppiAddC_32f_AC4R</a>      | 69 |
| 7.7.2.25 | <a href="#">nppiAddC_32f_C1IR</a>      | 69 |
| 7.7.2.26 | <a href="#">nppiAddC_32f_C1R</a>       | 69 |
| 7.7.2.27 | <a href="#">nppiAddC_32f_C3IR</a>      | 70 |
| 7.7.2.28 | <a href="#">nppiAddC_32f_C3R</a>       | 70 |
| 7.7.2.29 | <a href="#">nppiAddC_32f_C4IR</a>      | 70 |
| 7.7.2.30 | <a href="#">nppiAddC_32f_C4R</a>       | 71 |
| 7.7.2.31 | <a href="#">nppiAddC_32fc_AC4IR</a>    | 71 |
| 7.7.2.32 | <a href="#">nppiAddC_32fc_AC4R</a>     | 71 |
| 7.7.2.33 | <a href="#">nppiAddC_32fc_C1IR</a>     | 72 |
| 7.7.2.34 | <a href="#">nppiAddC_32fc_C1R</a>      | 72 |
| 7.7.2.35 | <a href="#">nppiAddC_32fc_C3IR</a>     | 72 |
| 7.7.2.36 | <a href="#">nppiAddC_32fc_C3R</a>      | 73 |
| 7.7.2.37 | <a href="#">nppiAddC_32fc_C4IR</a>     | 73 |
| 7.7.2.38 | <a href="#">nppiAddC_32fc_C4R</a>      | 73 |

|          |  |    |
|----------|--|----|
| 7.7.2.39 | <a href="#">nppiAddC_32s_C1IRSfs</a>   | 74 |
| 7.7.2.40 | <a href="#">nppiAddC_32s_C1RSfs</a>    | 74 |
| 7.7.2.41 | <a href="#">nppiAddC_32s_C3IRSfs</a>   | 74 |
| 7.7.2.42 | <a href="#">nppiAddC_32s_C3RSfs</a>    | 75 |
| 7.7.2.43 | <a href="#">nppiAddC_32sc_AC4IRSfs</a> | 75 |
| 7.7.2.44 | <a href="#">nppiAddC_32sc_AC4RSfs</a>  | 76 |
| 7.7.2.45 | <a href="#">nppiAddC_32sc_C1IRSfs</a>  | 76 |
| 7.7.2.46 | <a href="#">nppiAddC_32sc_C1RSfs</a>   | 76 |
| 7.7.2.47 | <a href="#">nppiAddC_32sc_C3IRSfs</a>  | 77 |
| 7.7.2.48 | <a href="#">nppiAddC_32sc_C3RSfs</a>   | 77 |
| 7.7.2.49 | <a href="#">nppiAddC_8u_AC4IRSfs</a>   | 78 |
| 7.7.2.50 | <a href="#">nppiAddC_8u_AC4RSfs</a>    | 78 |
| 7.7.2.51 | <a href="#">nppiAddC_8u_C1IRSfs</a>    | 78 |
| 7.7.2.52 | <a href="#">nppiAddC_8u_C1RSfs</a>     | 79 |
| 7.7.2.53 | <a href="#">nppiAddC_8u_C3IRSfs</a>    | 79 |
| 7.7.2.54 | <a href="#">nppiAddC_8u_C3RSfs</a>     | 79 |
| 7.7.2.55 | <a href="#">nppiAddC_8u_C4IRSfs</a>    | 80 |
| 7.7.2.56 | <a href="#">nppiAddC_8u_C4RSfs</a>     | 80 |
| 7.8      | <a href="#">MulC</a>                   | 81 |
| 7.8.1    | <a href="#">Detailed Description</a>   | 86 |
| 7.8.2    | <a href="#">Function Documentation</a> | 86 |
| 7.8.2.1  | <a href="#">nppiMulC_16s_AC4IRSfs</a>  | 86 |
| 7.8.2.2  | <a href="#">nppiMulC_16s_AC4RSfs</a>   | 86 |
| 7.8.2.3  | <a href="#">nppiMulC_16s_C1IRSfs</a>   | 87 |
| 7.8.2.4  | <a href="#">nppiMulC_16s_C1RSfs</a>    | 87 |
| 7.8.2.5  | <a href="#">nppiMulC_16s_C3IRSfs</a>   | 87 |
| 7.8.2.6  | <a href="#">nppiMulC_16s_C3RSfs</a>    | 88 |
| 7.8.2.7  | <a href="#">nppiMulC_16s_C4IRSfs</a>   | 88 |
| 7.8.2.8  | <a href="#">nppiMulC_16s_C4RSfs</a>    | 88 |
| 7.8.2.9  | <a href="#">nppiMulC_16sc_AC4IRSfs</a> | 89 |
| 7.8.2.10 | <a href="#">nppiMulC_16sc_AC4RSfs</a>  | 89 |
| 7.8.2.11 | <a href="#">nppiMulC_16sc_C1IRSfs</a>  | 90 |
| 7.8.2.12 | <a href="#">nppiMulC_16sc_C1RSfs</a>   | 90 |
| 7.8.2.13 | <a href="#">nppiMulC_16sc_C3IRSfs</a>  | 90 |
| 7.8.2.14 | <a href="#">nppiMulC_16sc_C3RSfs</a>   | 91 |
| 7.8.2.15 | <a href="#">nppiMulC_16u_AC4IRSfs</a>  | 91 |

|          |                                  |     |
|----------|----------------------------------|-----|
| 7.8.2.16 | nppiMulC_16u_AC4RSfs . . . . .   | 92  |
| 7.8.2.17 | nppiMulC_16u_C1IRSfs . . . . .   | 92  |
| 7.8.2.18 | nppiMulC_16u_C1RSfs . . . . .    | 92  |
| 7.8.2.19 | nppiMulC_16u_C3IRSfs . . . . .   | 93  |
| 7.8.2.20 | nppiMulC_16u_C3RSfs . . . . .    | 93  |
| 7.8.2.21 | nppiMulC_16u_C4IRSfs . . . . .   | 94  |
| 7.8.2.22 | nppiMulC_16u_C4RSfs . . . . .    | 94  |
| 7.8.2.23 | nppiMulC_32f_AC4IR . . . . .     | 94  |
| 7.8.2.24 | nppiMulC_32f_AC4R . . . . .      | 95  |
| 7.8.2.25 | nppiMulC_32f_C1IR . . . . .      | 95  |
| 7.8.2.26 | nppiMulC_32f_C1R . . . . .       | 95  |
| 7.8.2.27 | nppiMulC_32f_C3IR . . . . .      | 96  |
| 7.8.2.28 | nppiMulC_32f_C3R . . . . .       | 96  |
| 7.8.2.29 | nppiMulC_32f_C4IR . . . . .      | 96  |
| 7.8.2.30 | nppiMulC_32f_C4R . . . . .       | 97  |
| 7.8.2.31 | nppiMulC_32fc_AC4IR . . . . .    | 97  |
| 7.8.2.32 | nppiMulC_32fc_AC4R . . . . .     | 97  |
| 7.8.2.33 | nppiMulC_32fc_C1IR . . . . .     | 98  |
| 7.8.2.34 | nppiMulC_32fc_C1R . . . . .      | 98  |
| 7.8.2.35 | nppiMulC_32fc_C3IR . . . . .     | 98  |
| 7.8.2.36 | nppiMulC_32fc_C3R . . . . .      | 99  |
| 7.8.2.37 | nppiMulC_32fc_C4IR . . . . .     | 99  |
| 7.8.2.38 | nppiMulC_32fc_C4R . . . . .      | 99  |
| 7.8.2.39 | nppiMulC_32s_C1IRSfs . . . . .   | 100 |
| 7.8.2.40 | nppiMulC_32s_C1RSfs . . . . .    | 100 |
| 7.8.2.41 | nppiMulC_32s_C3IRSfs . . . . .   | 100 |
| 7.8.2.42 | nppiMulC_32s_C3RSfs . . . . .    | 101 |
| 7.8.2.43 | nppiMulC_32sc_AC4IRSfs . . . . . | 101 |
| 7.8.2.44 | nppiMulC_32sc_AC4RSfs . . . . .  | 102 |
| 7.8.2.45 | nppiMulC_32sc_C1IRSfs . . . . .  | 102 |
| 7.8.2.46 | nppiMulC_32sc_C1RSfs . . . . .   | 102 |
| 7.8.2.47 | nppiMulC_32sc_C3IRSfs . . . . .  | 103 |
| 7.8.2.48 | nppiMulC_32sc_C3RSfs . . . . .   | 103 |
| 7.8.2.49 | nppiMulC_8u_AC4IRSfs . . . . .   | 104 |
| 7.8.2.50 | nppiMulC_8u_AC4RSfs . . . . .    | 104 |
| 7.8.2.51 | nppiMulC_8u_C1IRSfs . . . . .    | 104 |

|          |                                      |     |
|----------|--------------------------------------|-----|
| 7.8.2.52 | <code>nppiMulC_8u_C1RSfs</code>      | 105 |
| 7.8.2.53 | <code>nppiMulC_8u_C3RSfs</code>      | 105 |
| 7.8.2.54 | <code>nppiMulC_8u_C3RSfs</code>      | 105 |
| 7.8.2.55 | <code>nppiMulC_8u_C4RSfs</code>      | 106 |
| 7.8.2.56 | <code>nppiMulC_8u_C4RSfs</code>      | 106 |
| 7.9      | <code>MulCScale</code>               | 107 |
| 7.9.1    | Detailed Description                 | 108 |
| 7.9.2    | Function Documentation               | 108 |
| 7.9.2.1  | <code>nppiMulCScale_16u_AC4IR</code> | 108 |
| 7.9.2.2  | <code>nppiMulCScale_16u_AC4R</code>  | 109 |
| 7.9.2.3  | <code>nppiMulCScale_16u_C1IR</code>  | 109 |
| 7.9.2.4  | <code>nppiMulCScale_16u_C1R</code>   | 109 |
| 7.9.2.5  | <code>nppiMulCScale_16u_C3IR</code>  | 110 |
| 7.9.2.6  | <code>nppiMulCScale_16u_C3R</code>   | 110 |
| 7.9.2.7  | <code>nppiMulCScale_16u_C4IR</code>  | 110 |
| 7.9.2.8  | <code>nppiMulCScale_16u_C4R</code>   | 111 |
| 7.9.2.9  | <code>nppiMulCScale_8u_AC4IR</code>  | 111 |
| 7.9.2.10 | <code>nppiMulCScale_8u_AC4R</code>   | 111 |
| 7.9.2.11 | <code>nppiMulCScale_8u_C1IR</code>   | 112 |
| 7.9.2.12 | <code>nppiMulCScale_8u_C1R</code>    | 112 |
| 7.9.2.13 | <code>nppiMulCScale_8u_C3IR</code>   | 112 |
| 7.9.2.14 | <code>nppiMulCScale_8u_C3R</code>    | 113 |
| 7.9.2.15 | <code>nppiMulCScale_8u_C4IR</code>   | 113 |
| 7.9.2.16 | <code>nppiMulCScale_8u_C4R</code>    | 113 |
| 7.10     | <code>SubC</code>                    | 114 |
| 7.10.1   | Detailed Description                 | 119 |
| 7.10.2   | Function Documentation               | 119 |
| 7.10.2.1 | <code>nppiSubC_16s_AC4IRSfs</code>   | 119 |
| 7.10.2.2 | <code>nppiSubC_16s_AC4RSfs</code>    | 119 |
| 7.10.2.3 | <code>nppiSubC_16s_C1IRSfs</code>    | 119 |
| 7.10.2.4 | <code>nppiSubC_16s_C1RSfs</code>     | 120 |
| 7.10.2.5 | <code>nppiSubC_16s_C3IRSfs</code>    | 120 |
| 7.10.2.6 | <code>nppiSubC_16s_C3RSfs</code>     | 121 |
| 7.10.2.7 | <code>nppiSubC_16s_C4IRSfs</code>    | 121 |
| 7.10.2.8 | <code>nppiSubC_16s_C4RSfs</code>     | 121 |
| 7.10.2.9 | <code>nppiSubC_16sc_AC4IRSfs</code>  | 122 |

|  |     |
|--|-----|
| 7.10.2.10 nppiSubC_16sc_AC4RSfs . . . . .  | 122 |
| 7.10.2.11 nppiSubC_16sc_C1IRSfs . . . . .  | 123 |
| 7.10.2.12 nppiSubC_16sc_C1RSfs . . . . .   | 123 |
| 7.10.2.13 nppiSubC_16sc_C3IRSfs . . . . .  | 123 |
| 7.10.2.14 nppiSubC_16sc_C3RSfs . . . . .   | 124 |
| 7.10.2.15 nppiSubC_16u_AC4IRSfs . . . . .  | 124 |
| 7.10.2.16 nppiSubC_16u_AC4RSfs . . . . .   | 125 |
| 7.10.2.17 nppiSubC_16u_C1IRSfs . . . . .   | 125 |
| 7.10.2.18 nppiSubC_16u_C1RSfs . . . . .    | 125 |
| 7.10.2.19 nppiSubC_16u_C3IRSfs . . . . .   | 126 |
| 7.10.2.20 nppiSubC_16u_C3RSfs . . . . .    | 126 |
| 7.10.2.21 nppiSubC_16u_C4IRSfs . . . . .   | 127 |
| 7.10.2.22 nppiSubC_16u_C4RSfs . . . . .    | 127 |
| 7.10.2.23 nppiSubC_32f_AC4IR . . . . .     | 127 |
| 7.10.2.24 nppiSubC_32f_AC4R . . . . .      | 128 |
| 7.10.2.25 nppiSubC_32f_C1IR . . . . .      | 128 |
| 7.10.2.26 nppiSubC_32f_C1R . . . . .       | 128 |
| 7.10.2.27 nppiSubC_32f_C3IR . . . . .      | 129 |
| 7.10.2.28 nppiSubC_32f_C3R . . . . .       | 129 |
| 7.10.2.29 nppiSubC_32f_C4IR . . . . .      | 129 |
| 7.10.2.30 nppiSubC_32f_C4R . . . . .       | 130 |
| 7.10.2.31 nppiSubC_32fc_AC4IR . . . . .    | 130 |
| 7.10.2.32 nppiSubC_32fc_AC4R . . . . .     | 130 |
| 7.10.2.33 nppiSubC_32fc_C1IR . . . . .     | 131 |
| 7.10.2.34 nppiSubC_32fc_C1R . . . . .      | 131 |
| 7.10.2.35 nppiSubC_32fc_C3IR . . . . .     | 131 |
| 7.10.2.36 nppiSubC_32fc_C3R . . . . .      | 132 |
| 7.10.2.37 nppiSubC_32fc_C4IR . . . . .     | 132 |
| 7.10.2.38 nppiSubC_32fc_C4R . . . . .      | 132 |
| 7.10.2.39 nppiSubC_32s_C1IRSfs . . . . .   | 133 |
| 7.10.2.40 nppiSubC_32s_C1RSfs . . . . .    | 133 |
| 7.10.2.41 nppiSubC_32s_C3IRSfs . . . . .   | 133 |
| 7.10.2.42 nppiSubC_32s_C3RSfs . . . . .    | 134 |
| 7.10.2.43 nppiSubC_32sc_AC4IRSfs . . . . . | 134 |
| 7.10.2.44 nppiSubC_32sc_AC4RSfs . . . . .  | 135 |
| 7.10.2.45 nppiSubC_32sc_C1IRSfs . . . . .  | 135 |

|           |                                     |     |
|-----------|-------------------------------------|-----|
| 7.10.2.46 | <code>nppiSubC_32sc_C1RSfs</code>   | 135 |
| 7.10.2.47 | <code>nppiSubC_32sc_C3IRSfs</code>  | 136 |
| 7.10.2.48 | <code>nppiSubC_32sc_C3RSfs</code>   | 136 |
| 7.10.2.49 | <code>nppiSubC_8u_AC4IRSfs</code>   | 137 |
| 7.10.2.50 | <code>nppiSubC_8u_AC4RSfs</code>    | 137 |
| 7.10.2.51 | <code>nppiSubC_8u_C1IRSfs</code>    | 137 |
| 7.10.2.52 | <code>nppiSubC_8u_C1RSfs</code>     | 138 |
| 7.10.2.53 | <code>nppiSubC_8u_C3IRSfs</code>    | 138 |
| 7.10.2.54 | <code>nppiSubC_8u_C3RSfs</code>     | 138 |
| 7.10.2.55 | <code>nppiSubC_8u_C4IRSfs</code>    | 139 |
| 7.10.2.56 | <code>nppiSubC_8u_C4RSfs</code>     | 139 |
| 7.11      | <code>DivC</code>                   | 140 |
| 7.11.1    | Detailed Description                | 145 |
| 7.11.2    | Function Documentation              | 145 |
| 7.11.2.1  | <code>nppiDivC_16s_AC4IRSfs</code>  | 145 |
| 7.11.2.2  | <code>nppiDivC_16s_AC4RSfs</code>   | 145 |
| 7.11.2.3  | <code>nppiDivC_16s_C1IRSfs</code>   | 146 |
| 7.11.2.4  | <code>nppiDivC_16s_C1RSfs</code>    | 146 |
| 7.11.2.5  | <code>nppiDivC_16s_C3IRSfs</code>   | 146 |
| 7.11.2.6  | <code>nppiDivC_16s_C3RSfs</code>    | 147 |
| 7.11.2.7  | <code>nppiDivC_16s_C4IRSfs</code>   | 147 |
| 7.11.2.8  | <code>nppiDivC_16s_C4RSfs</code>    | 147 |
| 7.11.2.9  | <code>nppiDivC_16sc_AC4IRSfs</code> | 148 |
| 7.11.2.10 | <code>nppiDivC_16sc_AC4RSfs</code>  | 148 |
| 7.11.2.11 | <code>nppiDivC_16sc_C1IRSfs</code>  | 149 |
| 7.11.2.12 | <code>nppiDivC_16sc_C1RSfs</code>   | 149 |
| 7.11.2.13 | <code>nppiDivC_16sc_C3IRSfs</code>  | 149 |
| 7.11.2.14 | <code>nppiDivC_16sc_C3RSfs</code>   | 150 |
| 7.11.2.15 | <code>nppiDivC_16u_AC4IRSfs</code>  | 150 |
| 7.11.2.16 | <code>nppiDivC_16u_AC4RSfs</code>   | 151 |
| 7.11.2.17 | <code>nppiDivC_16u_C1IRSfs</code>   | 151 |
| 7.11.2.18 | <code>nppiDivC_16u_C1RSfs</code>    | 151 |
| 7.11.2.19 | <code>nppiDivC_16u_C3IRSfs</code>   | 152 |
| 7.11.2.20 | <code>nppiDivC_16u_C3RSfs</code>    | 152 |
| 7.11.2.21 | <code>nppiDivC_16u_C4IRSfs</code>   | 153 |
| 7.11.2.22 | <code>nppiDivC_16u_C4RSfs</code>    | 153 |



|  |     |
|--|-----|
| 7.11.2.23 nppiDivC_32f_AC4IR . . . . .     | 153 |
| 7.11.2.24 nppiDivC_32f_AC4R . . . . .      | 154 |
| 7.11.2.25 nppiDivC_32f_C1IR . . . . .      | 154 |
| 7.11.2.26 nppiDivC_32f_C1R . . . . .       | 154 |
| 7.11.2.27 nppiDivC_32f_C3IR . . . . .      | 155 |
| 7.11.2.28 nppiDivC_32f_C3R . . . . .       | 155 |
| 7.11.2.29 nppiDivC_32f_C4IR . . . . .      | 155 |
| 7.11.2.30 nppiDivC_32f_C4R . . . . .       | 156 |
| 7.11.2.31 nppiDivC_32fc_AC4IR . . . . .    | 156 |
| 7.11.2.32 nppiDivC_32fc_AC4R . . . . .     | 156 |
| 7.11.2.33 nppiDivC_32fc_C1IR . . . . .     | 157 |
| 7.11.2.34 nppiDivC_32fc_C1R . . . . .      | 157 |
| 7.11.2.35 nppiDivC_32fc_C3IR . . . . .     | 157 |
| 7.11.2.36 nppiDivC_32fc_C3R . . . . .      | 158 |
| 7.11.2.37 nppiDivC_32fc_C4IR . . . . .     | 158 |
| 7.11.2.38 nppiDivC_32fc_C4R . . . . .      | 158 |
| 7.11.2.39 nppiDivC_32s_C1IRSfs . . . . .   | 159 |
| 7.11.2.40 nppiDivC_32s_C1RSfs . . . . .    | 159 |
| 7.11.2.41 nppiDivC_32s_C3IRSfs . . . . .   | 159 |
| 7.11.2.42 nppiDivC_32s_C3RSfs . . . . .    | 160 |
| 7.11.2.43 nppiDivC_32sc_AC4IRSfs . . . . . | 160 |
| 7.11.2.44 nppiDivC_32sc_AC4RSfs . . . . .  | 161 |
| 7.11.2.45 nppiDivC_32sc_C1IRSfs . . . . .  | 161 |
| 7.11.2.46 nppiDivC_32sc_C1RSfs . . . . .   | 161 |
| 7.11.2.47 nppiDivC_32sc_C3IRSfs . . . . .  | 162 |
| 7.11.2.48 nppiDivC_32sc_C3RSfs . . . . .   | 162 |
| 7.11.2.49 nppiDivC_8u_AC4IRSfs . . . . .   | 163 |
| 7.11.2.50 nppiDivC_8u_AC4RSfs . . . . .    | 163 |
| 7.11.2.51 nppiDivC_8u_C1IRSfs . . . . .    | 163 |
| 7.11.2.52 nppiDivC_8u_C1RSfs . . . . .     | 164 |
| 7.11.2.53 nppiDivC_8u_C3IRSfs . . . . .    | 164 |
| 7.11.2.54 nppiDivC_8u_C3RSfs . . . . .     | 164 |
| 7.11.2.55 nppiDivC_8u_C4IRSfs . . . . .    | 165 |
| 7.11.2.56 nppiDivC_8u_C4RSfs . . . . .     | 165 |
| 7.12 AbsDiffC . . . . .                    | 166 |
| 7.12.1 Detailed Description . . . . .      | 166 |

|           |                        |     |
|-----------|------------------------|-----|
| 7.12.2    | Function Documentation | 166 |
| 7.12.2.1  | npplAbsDiffC_16u_C1R   | 166 |
| 7.12.2.2  | npplAbsDiffC_32f_C1R   | 166 |
| 7.12.2.3  | npplAbsDiffC_8u_C1R    | 167 |
| 7.13      | Add                    | 168 |
| 7.13.1    | Detailed Description   | 173 |
| 7.13.2    | Function Documentation | 173 |
| 7.13.2.1  | npplAdd_16s_AC4IRSfs   | 173 |
| 7.13.2.2  | npplAdd_16s_AC4RSfs    | 173 |
| 7.13.2.3  | npplAdd_16s_C1IRSfs    | 174 |
| 7.13.2.4  | npplAdd_16s_C1RSfs     | 174 |
| 7.13.2.5  | npplAdd_16s_C3IRSfs    | 175 |
| 7.13.2.6  | npplAdd_16s_C3RSfs     | 175 |
| 7.13.2.7  | npplAdd_16s_C4IRSfs    | 176 |
| 7.13.2.8  | npplAdd_16s_C4RSfs     | 176 |
| 7.13.2.9  | npplAdd_16sc_AC4IRSfs  | 176 |
| 7.13.2.10 | npplAdd_16sc_AC4RSfs   | 177 |
| 7.13.2.11 | npplAdd_16sc_C1IRSfs   | 177 |
| 7.13.2.12 | npplAdd_16sc_C1RSfs    | 178 |
| 7.13.2.13 | npplAdd_16sc_C3IRSfs   | 178 |
| 7.13.2.14 | npplAdd_16sc_C3RSfs    | 178 |
| 7.13.2.15 | npplAdd_16u_AC4IRSfs   | 179 |
| 7.13.2.16 | npplAdd_16u_AC4RSfs    | 179 |
| 7.13.2.17 | npplAdd_16u_C1IRSfs    | 180 |
| 7.13.2.18 | npplAdd_16u_C1RSfs     | 180 |
| 7.13.2.19 | npplAdd_16u_C3IRSfs    | 181 |
| 7.13.2.20 | npplAdd_16u_C3RSfs     | 181 |
| 7.13.2.21 | npplAdd_16u_C4IRSfs    | 181 |
| 7.13.2.22 | npplAdd_16u_C4RSfs     | 182 |
| 7.13.2.23 | npplAdd_32f_AC4IR      | 182 |
| 7.13.2.24 | npplAdd_32f_AC4R       | 183 |
| 7.13.2.25 | npplAdd_32f_C1IR       | 183 |
| 7.13.2.26 | npplAdd_32f_C1R        | 183 |
| 7.13.2.27 | npplAdd_32f_C3IR       | 184 |
| 7.13.2.28 | npplAdd_32f_C3R        | 184 |
| 7.13.2.29 | npplAdd_32f_C4IR       | 185 |

|           |  |     |
|-----------|--|-----|
| 7.13.2.30 | <a href="#">nppiAdd_32f_C4R</a>            | 185 |
| 7.13.2.31 | <a href="#">nppiAdd_32fc_AC4IR</a>         | 185 |
| 7.13.2.32 | <a href="#">nppiAdd_32fc_AC4R</a>          | 186 |
| 7.13.2.33 | <a href="#">nppiAdd_32fc_C1IR</a>          | 186 |
| 7.13.2.34 | <a href="#">nppiAdd_32fc_C1R</a>           | 186 |
| 7.13.2.35 | <a href="#">nppiAdd_32fc_C3IR</a>          | 187 |
| 7.13.2.36 | <a href="#">nppiAdd_32fc_C3R</a>           | 187 |
| 7.13.2.37 | <a href="#">nppiAdd_32fc_C4IR</a>          | 188 |
| 7.13.2.38 | <a href="#">nppiAdd_32fc_C4R</a>           | 188 |
| 7.13.2.39 | <a href="#">nppiAdd_32s_C1IRSfs</a>        | 188 |
| 7.13.2.40 | <a href="#">nppiAdd_32s_C1R</a>            | 189 |
| 7.13.2.41 | <a href="#">nppiAdd_32s_C1RSfs</a>         | 189 |
| 7.13.2.42 | <a href="#">nppiAdd_32s_C3IRSfs</a>        | 190 |
| 7.13.2.43 | <a href="#">nppiAdd_32s_C3RSfs</a>         | 190 |
| 7.13.2.44 | <a href="#">nppiAdd_32sc_AC4IRSfs</a>      | 190 |
| 7.13.2.45 | <a href="#">nppiAdd_32sc_AC4RSfs</a>       | 191 |
| 7.13.2.46 | <a href="#">nppiAdd_32sc_C1IRSfs</a>       | 191 |
| 7.13.2.47 | <a href="#">nppiAdd_32sc_C1RSfs</a>        | 192 |
| 7.13.2.48 | <a href="#">nppiAdd_32sc_C3IRSfs</a>       | 192 |
| 7.13.2.49 | <a href="#">nppiAdd_32sc_C3RSfs</a>        | 192 |
| 7.13.2.50 | <a href="#">nppiAdd_8u_AC4IRSfs</a>        | 193 |
| 7.13.2.51 | <a href="#">nppiAdd_8u_AC4RSfs</a>         | 193 |
| 7.13.2.52 | <a href="#">nppiAdd_8u_C1IRSfs</a>         | 194 |
| 7.13.2.53 | <a href="#">nppiAdd_8u_C1RSfs</a>          | 194 |
| 7.13.2.54 | <a href="#">nppiAdd_8u_C3IRSfs</a>         | 195 |
| 7.13.2.55 | <a href="#">nppiAdd_8u_C3RSfs</a>          | 195 |
| 7.13.2.56 | <a href="#">nppiAdd_8u_C4IRSfs</a>         | 195 |
| 7.13.2.57 | <a href="#">nppiAdd_8u_C4RSfs</a>          | 196 |
| 7.14      | <a href="#">AddSquare</a>                  | 197 |
| 7.14.1    | <a href="#">Detailed Description</a>       | 197 |
| 7.14.2    | <a href="#">Function Documentation</a>     | 197 |
| 7.14.2.1  | <a href="#">nppiAddSquare_16u32f_C1IMR</a> | 197 |
| 7.14.2.2  | <a href="#">nppiAddSquare_16u32f_C1IR</a>  | 198 |
| 7.14.2.3  | <a href="#">nppiAddSquare_32f_C1IMR</a>    | 198 |
| 7.14.2.4  | <a href="#">nppiAddSquare_32f_C1IR</a>     | 199 |
| 7.14.2.5  | <a href="#">nppiAddSquare_8u32f_C1IMR</a>  | 199 |

|           |   |     |
|-----------|---|-----|
| 7.14.2.6  | <code>npplAddSquare_8u32f_C1IR</code>     | 199 |
| 7.15      | <code>AddProduct</code>                   | 200 |
| 7.15.1    | Detailed Description                      | 200 |
| 7.15.2    | Function Documentation                    | 200 |
| 7.15.2.1  | <code>npplAddProduct_16u32f_C1IMR</code>  | 200 |
| 7.15.2.2  | <code>npplAddProduct_16u32f_C1IR</code>   | 201 |
| 7.15.2.3  | <code>npplAddProduct_32f_C1IMR</code>     | 201 |
| 7.15.2.4  | <code>npplAddProduct_32f_C1IR</code>      | 202 |
| 7.15.2.5  | <code>npplAddProduct_8u32f_C1IMR</code>   | 202 |
| 7.15.2.6  | <code>npplAddProduct_8u32f_C1IR</code>    | 203 |
| 7.16      | <code>AddWeighted</code>                  | 204 |
| 7.16.1    | Detailed Description                      | 204 |
| 7.16.2    | Function Documentation                    | 204 |
| 7.16.2.1  | <code>npplAddWeighted_16u32f_C1IMR</code> | 204 |
| 7.16.2.2  | <code>npplAddWeighted_16u32f_C1IR</code>  | 205 |
| 7.16.2.3  | <code>npplAddWeighted_32f_C1IMR</code>    | 205 |
| 7.16.2.4  | <code>npplAddWeighted_32f_C1IR</code>     | 206 |
| 7.16.2.5  | <code>npplAddWeighted_8u32f_C1IMR</code>  | 206 |
| 7.16.2.6  | <code>npplAddWeighted_8u32f_C1IR</code>   | 207 |
| 7.17      | <code>Mul</code>                          | 208 |
| 7.17.1    | Detailed Description                      | 213 |
| 7.17.2    | Function Documentation                    | 213 |
| 7.17.2.1  | <code>npplMul_16s_AC4IRSfs</code>         | 213 |
| 7.17.2.2  | <code>npplMul_16s_AC4RSfs</code>          | 214 |
| 7.17.2.3  | <code>npplMul_16s_C1IRSfs</code>          | 214 |
| 7.17.2.4  | <code>npplMul_16s_C1RSfs</code>           | 214 |
| 7.17.2.5  | <code>npplMul_16s_C3IRSfs</code>          | 215 |
| 7.17.2.6  | <code>npplMul_16s_C3RSfs</code>           | 215 |
| 7.17.2.7  | <code>npplMul_16s_C4IRSfs</code>          | 216 |
| 7.17.2.8  | <code>npplMul_16s_C4RSfs</code>           | 216 |
| 7.17.2.9  | <code>npplMul_16sc_AC4IRSfs</code>        | 216 |
| 7.17.2.10 | <code>npplMul_16sc_AC4RSfs</code>         | 217 |
| 7.17.2.11 | <code>npplMul_16sc_C1IRSfs</code>         | 217 |
| 7.17.2.12 | <code>npplMul_16sc_C1RSfs</code>          | 218 |
| 7.17.2.13 | <code>npplMul_16sc_C3IRSfs</code>         | 218 |
| 7.17.2.14 | <code>npplMul_16sc_C3RSfs</code>          | 218 |

|   |     |
|---|-----|
| 7.17.2.15 nppiMul_16u_AC4IRSfs . . . . .  | 219 |
| 7.17.2.16 nppiMul_16u_AC4RSfs . . . . .   | 219 |
| 7.17.2.17 nppiMul_16u_C1IRSfs . . . . .   | 220 |
| 7.17.2.18 nppiMul_16u_C1RSfs . . . . .    | 220 |
| 7.17.2.19 nppiMul_16u_C3IRSfs . . . . .   | 221 |
| 7.17.2.20 nppiMul_16u_C3RSfs . . . . .    | 221 |
| 7.17.2.21 nppiMul_16u_C4IRSfs . . . . .   | 221 |
| 7.17.2.22 nppiMul_16u_C4RSfs . . . . .    | 222 |
| 7.17.2.23 nppiMul_32f_AC4IR . . . . .     | 222 |
| 7.17.2.24 nppiMul_32f_AC4R . . . . .      | 223 |
| 7.17.2.25 nppiMul_32f_C1IR . . . . .      | 223 |
| 7.17.2.26 nppiMul_32f_C1R . . . . .       | 223 |
| 7.17.2.27 nppiMul_32f_C3IR . . . . .      | 224 |
| 7.17.2.28 nppiMul_32f_C3R . . . . .       | 224 |
| 7.17.2.29 nppiMul_32f_C4IR . . . . .      | 225 |
| 7.17.2.30 nppiMul_32f_C4R . . . . .       | 225 |
| 7.17.2.31 nppiMul_32fc_AC4IR . . . . .    | 225 |
| 7.17.2.32 nppiMul_32fc_AC4R . . . . .     | 226 |
| 7.17.2.33 nppiMul_32fc_C1IR . . . . .     | 226 |
| 7.17.2.34 nppiMul_32fc_C1R . . . . .      | 226 |
| 7.17.2.35 nppiMul_32fc_C3IR . . . . .     | 227 |
| 7.17.2.36 nppiMul_32fc_C3R . . . . .      | 227 |
| 7.17.2.37 nppiMul_32fc_C4IR . . . . .     | 228 |
| 7.17.2.38 nppiMul_32fc_C4R . . . . .      | 228 |
| 7.17.2.39 nppiMul_32s_C1IRSfs . . . . .   | 228 |
| 7.17.2.40 nppiMul_32s_C1R . . . . .       | 229 |
| 7.17.2.41 nppiMul_32s_C1RSfs . . . . .    | 229 |
| 7.17.2.42 nppiMul_32s_C3IRSfs . . . . .   | 230 |
| 7.17.2.43 nppiMul_32s_C3RSfs . . . . .    | 230 |
| 7.17.2.44 nppiMul_32sc_AC4IRSfs . . . . . | 230 |
| 7.17.2.45 nppiMul_32sc_AC4RSfs . . . . .  | 231 |
| 7.17.2.46 nppiMul_32sc_C1IRSfs . . . . .  | 231 |
| 7.17.2.47 nppiMul_32sc_C1RSfs . . . . .   | 232 |
| 7.17.2.48 nppiMul_32sc_C3IRSfs . . . . .  | 232 |
| 7.17.2.49 nppiMul_32sc_C3RSfs . . . . .   | 232 |
| 7.17.2.50 nppiMul_8u_AC4IRSfs . . . . .   | 233 |

|           |                                     |     |
|-----------|-------------------------------------|-----|
| 7.17.2.51 | <code>nppiMul_8u_AC4RSfs</code>     | 233 |
| 7.17.2.52 | <code>nppiMul_8u_C1RSfs</code>      | 234 |
| 7.17.2.53 | <code>nppiMul_8u_C1RSfs</code>      | 234 |
| 7.17.2.54 | <code>nppiMul_8u_C3RSfs</code>      | 235 |
| 7.17.2.55 | <code>nppiMul_8u_C3RSfs</code>      | 235 |
| 7.17.2.56 | <code>nppiMul_8u_C4RSfs</code>      | 235 |
| 7.17.2.57 | <code>nppiMul_8u_C4RSfs</code>      | 236 |
| 7.18      | <b>MulScale</b>                     | 237 |
| 7.18.1    | Detailed Description                | 238 |
| 7.18.2    | Function Documentation              | 238 |
| 7.18.2.1  | <code>nppiMulScale_16u_AC4IR</code> | 238 |
| 7.18.2.2  | <code>nppiMulScale_16u_AC4R</code>  | 239 |
| 7.18.2.3  | <code>nppiMulScale_16u_C1IR</code>  | 239 |
| 7.18.2.4  | <code>nppiMulScale_16u_C1R</code>   | 240 |
| 7.18.2.5  | <code>nppiMulScale_16u_C3IR</code>  | 240 |
| 7.18.2.6  | <code>nppiMulScale_16u_C3R</code>   | 240 |
| 7.18.2.7  | <code>nppiMulScale_16u_C4IR</code>  | 241 |
| 7.18.2.8  | <code>nppiMulScale_16u_C4R</code>   | 241 |
| 7.18.2.9  | <code>nppiMulScale_8u_AC4IR</code>  | 242 |
| 7.18.2.10 | <code>nppiMulScale_8u_AC4R</code>   | 242 |
| 7.18.2.11 | <code>nppiMulScale_8u_C1IR</code>   | 242 |
| 7.18.2.12 | <code>nppiMulScale_8u_C1R</code>    | 243 |
| 7.18.2.13 | <code>nppiMulScale_8u_C3IR</code>   | 243 |
| 7.18.2.14 | <code>nppiMulScale_8u_C3R</code>    | 244 |
| 7.18.2.15 | <code>nppiMulScale_8u_C4IR</code>   | 244 |
| 7.18.2.16 | <code>nppiMulScale_8u_C4R</code>    | 244 |
| 7.19      | <b>Sub</b>                          | 246 |
| 7.19.1    | Detailed Description                | 251 |
| 7.19.2    | Function Documentation              | 251 |
| 7.19.2.1  | <code>nppiSub_16s_AC4IRSfs</code>   | 251 |
| 7.19.2.2  | <code>nppiSub_16s_AC4RSfs</code>    | 252 |
| 7.19.2.3  | <code>nppiSub_16s_C1IRSfs</code>    | 252 |
| 7.19.2.4  | <code>nppiSub_16s_C1RSfs</code>     | 253 |
| 7.19.2.5  | <code>nppiSub_16s_C3IRSfs</code>    | 253 |
| 7.19.2.6  | <code>nppiSub_16s_C3RSfs</code>     | 253 |
| 7.19.2.7  | <code>nppiSub_16s_C4IRSfs</code>    | 254 |

|           |                       |     |
|-----------|-----------------------|-----|
| 7.19.2.8  | nppiSub_16s_C4RSfs    | 254 |
| 7.19.2.9  | nppiSub_16sc_AC4IRSfs | 255 |
| 7.19.2.10 | nppiSub_16sc_AC4RSfs  | 255 |
| 7.19.2.11 | nppiSub_16sc_C1IRSfs  | 255 |
| 7.19.2.12 | nppiSub_16sc_C1RSfs   | 256 |
| 7.19.2.13 | nppiSub_16sc_C3IRSfs  | 256 |
| 7.19.2.14 | nppiSub_16sc_C3RSfs   | 257 |
| 7.19.2.15 | nppiSub_16u_AC4IRSfs  | 257 |
| 7.19.2.16 | nppiSub_16u_AC4RSfs   | 257 |
| 7.19.2.17 | nppiSub_16u_C1IRSfs   | 258 |
| 7.19.2.18 | nppiSub_16u_C1RSfs    | 258 |
| 7.19.2.19 | nppiSub_16u_C3IRSfs   | 259 |
| 7.19.2.20 | nppiSub_16u_C3RSfs    | 259 |
| 7.19.2.21 | nppiSub_16u_C4IRSfs   | 260 |
| 7.19.2.22 | nppiSub_16u_C4RSfs    | 260 |
| 7.19.2.23 | nppiSub_32f_AC4IR     | 260 |
| 7.19.2.24 | nppiSub_32f_AC4R      | 261 |
| 7.19.2.25 | nppiSub_32f_C1IR      | 261 |
| 7.19.2.26 | nppiSub_32f_C1R       | 262 |
| 7.19.2.27 | nppiSub_32f_C3IR      | 262 |
| 7.19.2.28 | nppiSub_32f_C3R       | 262 |
| 7.19.2.29 | nppiSub_32f_C4IR      | 263 |
| 7.19.2.30 | nppiSub_32f_C4R       | 263 |
| 7.19.2.31 | nppiSub_32fc_AC4IR    | 264 |
| 7.19.2.32 | nppiSub_32fc_AC4R     | 264 |
| 7.19.2.33 | nppiSub_32fc_C1IR     | 264 |
| 7.19.2.34 | nppiSub_32fc_C1R      | 265 |
| 7.19.2.35 | nppiSub_32fc_C3IR     | 265 |
| 7.19.2.36 | nppiSub_32fc_C3R      | 266 |
| 7.19.2.37 | nppiSub_32fc_C4IR     | 266 |
| 7.19.2.38 | nppiSub_32fc_C4R      | 266 |
| 7.19.2.39 | nppiSub_32s_C1IRSfs   | 267 |
| 7.19.2.40 | nppiSub_32s_C1R       | 267 |
| 7.19.2.41 | nppiSub_32s_C1RSfs    | 268 |
| 7.19.2.42 | nppiSub_32s_C3IRSfs   | 268 |
| 7.19.2.43 | nppiSub_32s_C3RSfs    | 268 |

|           |  |     |
|-----------|--|-----|
| 7.19.2.44 | <a href="#">nppiSub_32s_C4IRSfs</a>    | 269 |
| 7.19.2.45 | <a href="#">nppiSub_32s_C4RSfs</a>     | 269 |
| 7.19.2.46 | <a href="#">nppiSub_32sc_AC4IRSfs</a>  | 270 |
| 7.19.2.47 | <a href="#">nppiSub_32sc_AC4RSfs</a>   | 270 |
| 7.19.2.48 | <a href="#">nppiSub_32sc_C1IRSfs</a>   | 271 |
| 7.19.2.49 | <a href="#">nppiSub_32sc_C1RSfs</a>    | 271 |
| 7.19.2.50 | <a href="#">nppiSub_32sc_C3IRSfs</a>   | 271 |
| 7.19.2.51 | <a href="#">nppiSub_32sc_C3RSfs</a>    | 272 |
| 7.19.2.52 | <a href="#">nppiSub_8u_AC4IRSfs</a>    | 272 |
| 7.19.2.53 | <a href="#">nppiSub_8u_AC4RSfs</a>     | 273 |
| 7.19.2.54 | <a href="#">nppiSub_8u_C1IRSfs</a>     | 273 |
| 7.19.2.55 | <a href="#">nppiSub_8u_C1RSfs</a>      | 273 |
| 7.19.2.56 | <a href="#">nppiSub_8u_C3IRSfs</a>     | 274 |
| 7.19.2.57 | <a href="#">nppiSub_8u_C3RSfs</a>      | 274 |
| 7.19.2.58 | <a href="#">nppiSub_8u_C4IRSfs</a>     | 275 |
| 7.19.2.59 | <a href="#">nppiSub_8u_C4RSfs</a>      | 275 |
| 7.20      | <a href="#">Div</a>                    | 276 |
| 7.20.1    | <a href="#">Detailed Description</a>   | 281 |
| 7.20.2    | <a href="#">Function Documentation</a> | 281 |
| 7.20.2.1  | <a href="#">nppiDiv_16s_AC4IRSfs</a>   | 281 |
| 7.20.2.2  | <a href="#">nppiDiv_16s_AC4RSfs</a>    | 281 |
| 7.20.2.3  | <a href="#">nppiDiv_16s_C1IRSfs</a>    | 282 |
| 7.20.2.4  | <a href="#">nppiDiv_16s_C1RSfs</a>     | 282 |
| 7.20.2.5  | <a href="#">nppiDiv_16s_C3IRSfs</a>    | 283 |
| 7.20.2.6  | <a href="#">nppiDiv_16s_C3RSfs</a>     | 283 |
| 7.20.2.7  | <a href="#">nppiDiv_16s_C4IRSfs</a>    | 283 |
| 7.20.2.8  | <a href="#">nppiDiv_16s_C4RSfs</a>     | 284 |
| 7.20.2.9  | <a href="#">nppiDiv_16sc_AC4IRSfs</a>  | 284 |
| 7.20.2.10 | <a href="#">nppiDiv_16sc_AC4RSfs</a>   | 285 |
| 7.20.2.11 | <a href="#">nppiDiv_16sc_C1IRSfs</a>   | 285 |
| 7.20.2.12 | <a href="#">nppiDiv_16sc_C1RSfs</a>    | 285 |
| 7.20.2.13 | <a href="#">nppiDiv_16sc_C3IRSfs</a>   | 286 |
| 7.20.2.14 | <a href="#">nppiDiv_16sc_C3RSfs</a>    | 286 |
| 7.20.2.15 | <a href="#">nppiDiv_16u_AC4IRSfs</a>   | 287 |
| 7.20.2.16 | <a href="#">nppiDiv_16u_AC4RSfs</a>    | 287 |
| 7.20.2.17 | <a href="#">nppiDiv_16u_C1IRSfs</a>    | 288 |



|   |     |
|---|-----|
| 7.20.2.18 nppiDiv_16u_C1RSfs . . . . .    | 288 |
| 7.20.2.19 nppiDiv_16u_C3IRSfs . . . . .   | 288 |
| 7.20.2.20 nppiDiv_16u_C3RSfs . . . . .    | 289 |
| 7.20.2.21 nppiDiv_16u_C4IRSfs . . . . .   | 289 |
| 7.20.2.22 nppiDiv_16u_C4RSfs . . . . .    | 290 |
| 7.20.2.23 nppiDiv_32f_AC4IR . . . . .     | 290 |
| 7.20.2.24 nppiDiv_32f_AC4R . . . . .      | 290 |
| 7.20.2.25 nppiDiv_32f_C1IR . . . . .      | 291 |
| 7.20.2.26 nppiDiv_32f_C1R . . . . .       | 291 |
| 7.20.2.27 nppiDiv_32f_C3IR . . . . .      | 292 |
| 7.20.2.28 nppiDiv_32f_C3R . . . . .       | 292 |
| 7.20.2.29 nppiDiv_32f_C4IR . . . . .      | 292 |
| 7.20.2.30 nppiDiv_32f_C4R . . . . .       | 293 |
| 7.20.2.31 nppiDiv_32fc_AC4IR . . . . .    | 293 |
| 7.20.2.32 nppiDiv_32fc_AC4R . . . . .     | 293 |
| 7.20.2.33 nppiDiv_32fc_C1IR . . . . .     | 294 |
| 7.20.2.34 nppiDiv_32fc_C1R . . . . .      | 294 |
| 7.20.2.35 nppiDiv_32fc_C3IR . . . . .     | 295 |
| 7.20.2.36 nppiDiv_32fc_C3R . . . . .      | 295 |
| 7.20.2.37 nppiDiv_32fc_C4IR . . . . .     | 295 |
| 7.20.2.38 nppiDiv_32fc_C4R . . . . .      | 296 |
| 7.20.2.39 nppiDiv_32s_C1IRSfs . . . . .   | 296 |
| 7.20.2.40 nppiDiv_32s_C1R . . . . .       | 296 |
| 7.20.2.41 nppiDiv_32s_C1RSfs . . . . .    | 297 |
| 7.20.2.42 nppiDiv_32s_C3IRSfs . . . . .   | 297 |
| 7.20.2.43 nppiDiv_32s_C3RSfs . . . . .    | 298 |
| 7.20.2.44 nppiDiv_32sc_AC4IRSfs . . . . . | 298 |
| 7.20.2.45 nppiDiv_32sc_AC4RSfs . . . . .  | 298 |
| 7.20.2.46 nppiDiv_32sc_C1IRSfs . . . . .  | 299 |
| 7.20.2.47 nppiDiv_32sc_C1RSfs . . . . .   | 299 |
| 7.20.2.48 nppiDiv_32sc_C3IRSfs . . . . .  | 300 |
| 7.20.2.49 nppiDiv_32sc_C3RSfs . . . . .   | 300 |
| 7.20.2.50 nppiDiv_8u_AC4IRSfs . . . . .   | 301 |
| 7.20.2.51 nppiDiv_8u_AC4RSfs . . . . .    | 301 |
| 7.20.2.52 nppiDiv_8u_C1IRSfs . . . . .    | 301 |
| 7.20.2.53 nppiDiv_8u_C1RSfs . . . . .     | 302 |

|           |   |     |
|-----------|---|-----|
| 7.20.2.54 | <code>nppiDiv_8u_C3IRSfs</code>         | 302 |
| 7.20.2.55 | <code>nppiDiv_8u_C3RSfs</code>          | 303 |
| 7.20.2.56 | <code>nppiDiv_8u_C4IRSfs</code>         | 303 |
| 7.20.2.57 | <code>nppiDiv_8u_C4RSfs</code>          | 303 |
| 7.21      | <code>Div_Round</code>                  | 305 |
| 7.21.1    | Detailed Description                    | 307 |
| 7.21.2    | Function Documentation                  | 307 |
| 7.21.2.1  | <code>nppiDiv_Round_16s_AC4IRSfs</code> | 307 |
| 7.21.2.2  | <code>nppiDiv_Round_16s_AC4RSfs</code>  | 308 |
| 7.21.2.3  | <code>nppiDiv_Round_16s_C1IRSfs</code>  | 308 |
| 7.21.2.4  | <code>nppiDiv_Round_16s_C1RSfs</code>   | 309 |
| 7.21.2.5  | <code>nppiDiv_Round_16s_C3IRSfs</code>  | 309 |
| 7.21.2.6  | <code>nppiDiv_Round_16s_C3RSfs</code>   | 310 |
| 7.21.2.7  | <code>nppiDiv_Round_16s_C4IRSfs</code>  | 310 |
| 7.21.2.8  | <code>nppiDiv_Round_16s_C4RSfs</code>   | 311 |
| 7.21.2.9  | <code>nppiDiv_Round_16u_AC4IRSfs</code> | 311 |
| 7.21.2.10 | <code>nppiDiv_Round_16u_AC4RSfs</code>  | 312 |
| 7.21.2.11 | <code>nppiDiv_Round_16u_C1IRSfs</code>  | 312 |
| 7.21.2.12 | <code>nppiDiv_Round_16u_C1RSfs</code>   | 313 |
| 7.21.2.13 | <code>nppiDiv_Round_16u_C3IRSfs</code>  | 313 |
| 7.21.2.14 | <code>nppiDiv_Round_16u_C3RSfs</code>   | 314 |
| 7.21.2.15 | <code>nppiDiv_Round_16u_C4IRSfs</code>  | 314 |
| 7.21.2.16 | <code>nppiDiv_Round_16u_C4RSfs</code>   | 315 |
| 7.21.2.17 | <code>nppiDiv_Round_8u_AC4IRSfs</code>  | 315 |
| 7.21.2.18 | <code>nppiDiv_Round_8u_AC4RSfs</code>   | 316 |
| 7.21.2.19 | <code>nppiDiv_Round_8u_C1IRSfs</code>   | 316 |
| 7.21.2.20 | <code>nppiDiv_Round_8u_C1RSfs</code>    | 317 |
| 7.21.2.21 | <code>nppiDiv_Round_8u_C3IRSfs</code>   | 317 |
| 7.21.2.22 | <code>nppiDiv_Round_8u_C3RSfs</code>    | 318 |
| 7.21.2.23 | <code>nppiDiv_Round_8u_C4IRSfs</code>   | 318 |
| 7.21.2.24 | <code>nppiDiv_Round_8u_C4RSfs</code>    | 319 |
| 7.22      | <code>Abs</code>                        | 320 |
| 7.22.1    | Detailed Description                    | 321 |
| 7.22.2    | Function Documentation                  | 321 |
| 7.22.2.1  | <code>nppiAbs_16s_AC4IR</code>          | 321 |
| 7.22.2.2  | <code>nppiAbs_16s_AC4R</code>           | 321 |

|           |  |     |
|-----------|--|-----|
| 7.22.2.3  | <a href="#">nppiAbs_16s_C1IR</a>       | 322 |
| 7.22.2.4  | <a href="#">nppiAbs_16s_C1R</a>        | 322 |
| 7.22.2.5  | <a href="#">nppiAbs_16s_C3IR</a>       | 322 |
| 7.22.2.6  | <a href="#">nppiAbs_16s_C3R</a>        | 323 |
| 7.22.2.7  | <a href="#">nppiAbs_16s_C4IR</a>       | 323 |
| 7.22.2.8  | <a href="#">nppiAbs_16s_C4R</a>        | 323 |
| 7.22.2.9  | <a href="#">nppiAbs_32f_AC4IR</a>      | 324 |
| 7.22.2.10 | <a href="#">nppiAbs_32f_AC4R</a>       | 324 |
| 7.22.2.11 | <a href="#">nppiAbs_32f_C1IR</a>       | 324 |
| 7.22.2.12 | <a href="#">nppiAbs_32f_C1R</a>        | 325 |
| 7.22.2.13 | <a href="#">nppiAbs_32f_C3IR</a>       | 325 |
| 7.22.2.14 | <a href="#">nppiAbs_32f_C3R</a>        | 325 |
| 7.22.2.15 | <a href="#">nppiAbs_32f_C4IR</a>       | 326 |
| 7.22.2.16 | <a href="#">nppiAbs_32f_C4R</a>        | 326 |
| 7.23      | <a href="#">AbsDiff</a>                | 327 |
| 7.23.1    | <a href="#">Detailed Description</a>   | 327 |
| 7.23.2    | <a href="#">Function Documentation</a> | 327 |
| 7.23.2.1  | <a href="#">nppiAbsDiff_16u_C1R</a>    | 327 |
| 7.23.2.2  | <a href="#">nppiAbsDiff_32f_C1R</a>    | 328 |
| 7.23.2.3  | <a href="#">nppiAbsDiff_8u_C1R</a>     | 328 |
| 7.23.2.4  | <a href="#">nppiAbsDiff_8u_C3R</a>     | 328 |
| 7.23.2.5  | <a href="#">nppiAbsDiff_8u_C4R</a>     | 329 |
| 7.24      | <a href="#">Sqr</a>                    | 330 |
| 7.24.1    | <a href="#">Detailed Description</a>   | 332 |
| 7.24.2    | <a href="#">Function Documentation</a> | 333 |
| 7.24.2.1  | <a href="#">nppiSqr_16s_AC4IRSfs</a>   | 333 |
| 7.24.2.2  | <a href="#">nppiSqr_16s_AC4RSfs</a>    | 333 |
| 7.24.2.3  | <a href="#">nppiSqr_16s_C1IRSfs</a>    | 333 |
| 7.24.2.4  | <a href="#">nppiSqr_16s_C1RSfs</a>     | 334 |
| 7.24.2.5  | <a href="#">nppiSqr_16s_C3IRSfs</a>    | 334 |
| 7.24.2.6  | <a href="#">nppiSqr_16s_C3RSfs</a>     | 334 |
| 7.24.2.7  | <a href="#">nppiSqr_16s_C4IRSfs</a>    | 335 |
| 7.24.2.8  | <a href="#">nppiSqr_16s_C4RSfs</a>     | 335 |
| 7.24.2.9  | <a href="#">nppiSqr_16u_AC4IRSfs</a>   | 335 |
| 7.24.2.10 | <a href="#">nppiSqr_16u_AC4RSfs</a>    | 336 |
| 7.24.2.11 | <a href="#">nppiSqr_16u_C1IRSfs</a>    | 336 |

|           |                                    |     |
|-----------|------------------------------------|-----|
| 7.24.2.12 | <code>nppiSqr_16u_C1RSfs</code>    | 336 |
| 7.24.2.13 | <code>nppiSqr_16u_C3IRSfs</code>   | 337 |
| 7.24.2.14 | <code>nppiSqr_16u_C3RSfs</code>    | 337 |
| 7.24.2.15 | <code>nppiSqr_16u_C4IRSfs</code>   | 337 |
| 7.24.2.16 | <code>nppiSqr_16u_C4RSfs</code>    | 338 |
| 7.24.2.17 | <code>nppiSqr_32f_AC4IR</code>     | 338 |
| 7.24.2.18 | <code>nppiSqr_32f_AC4R</code>      | 338 |
| 7.24.2.19 | <code>nppiSqr_32f_C1IR</code>      | 339 |
| 7.24.2.20 | <code>nppiSqr_32f_C1R</code>       | 339 |
| 7.24.2.21 | <code>nppiSqr_32f_C3IR</code>      | 339 |
| 7.24.2.22 | <code>nppiSqr_32f_C3R</code>       | 340 |
| 7.24.2.23 | <code>nppiSqr_32f_C4IR</code>      | 340 |
| 7.24.2.24 | <code>nppiSqr_32f_C4R</code>       | 340 |
| 7.24.2.25 | <code>nppiSqr_8u_AC4IRSfs</code>   | 341 |
| 7.24.2.26 | <code>nppiSqr_8u_AC4RSfs</code>    | 341 |
| 7.24.2.27 | <code>nppiSqr_8u_C1IRSfs</code>    | 341 |
| 7.24.2.28 | <code>nppiSqr_8u_C1RSfs</code>     | 342 |
| 7.24.2.29 | <code>nppiSqr_8u_C3IRSfs</code>    | 342 |
| 7.24.2.30 | <code>nppiSqr_8u_C3RSfs</code>     | 342 |
| 7.24.2.31 | <code>nppiSqr_8u_C4IRSfs</code>    | 343 |
| 7.24.2.32 | <code>nppiSqr_8u_C4RSfs</code>     | 343 |
| 7.25      | <code>Sqrt</code>                  | 344 |
| 7.25.1    | Detailed Description               | 346 |
| 7.25.2    | Function Documentation             | 346 |
| 7.25.2.1  | <code>nppiSqrt_16s_AC4IRSfs</code> | 346 |
| 7.25.2.2  | <code>nppiSqrt_16s_AC4RSfs</code>  | 347 |
| 7.25.2.3  | <code>nppiSqrt_16s_C1IRSfs</code>  | 347 |
| 7.25.2.4  | <code>nppiSqrt_16s_C1RSfs</code>   | 347 |
| 7.25.2.5  | <code>nppiSqrt_16s_C3IRSfs</code>  | 348 |
| 7.25.2.6  | <code>nppiSqrt_16s_C3RSfs</code>   | 348 |
| 7.25.2.7  | <code>nppiSqrt_16u_AC4IRSfs</code> | 348 |
| 7.25.2.8  | <code>nppiSqrt_16u_AC4RSfs</code>  | 349 |
| 7.25.2.9  | <code>nppiSqrt_16u_C1IRSfs</code>  | 349 |
| 7.25.2.10 | <code>nppiSqrt_16u_C1RSfs</code>   | 350 |
| 7.25.2.11 | <code>nppiSqrt_16u_C3IRSfs</code>  | 350 |
| 7.25.2.12 | <code>nppiSqrt_16u_C3RSfs</code>   | 350 |

|           |  |     |
|-----------|--|-----|
| 7.25.2.13 | <a href="#">nppiSqrt_32f_AC4IR</a>     | 351 |
| 7.25.2.14 | <a href="#">nppiSqrt_32f_AC4R</a>      | 351 |
| 7.25.2.15 | <a href="#">nppiSqrt_32f_C1IR</a>      | 351 |
| 7.25.2.16 | <a href="#">nppiSqrt_32f_C1R</a>       | 352 |
| 7.25.2.17 | <a href="#">nppiSqrt_32f_C3IR</a>      | 352 |
| 7.25.2.18 | <a href="#">nppiSqrt_32f_C3R</a>       | 352 |
| 7.25.2.19 | <a href="#">nppiSqrt_32f_C4IR</a>      | 353 |
| 7.25.2.20 | <a href="#">nppiSqrt_32f_C4R</a>       | 353 |
| 7.25.2.21 | <a href="#">nppiSqrt_8u_AC4IRSfs</a>   | 353 |
| 7.25.2.22 | <a href="#">nppiSqrt_8u_AC4RSfs</a>    | 354 |
| 7.25.2.23 | <a href="#">nppiSqrt_8u_C1IRSfs</a>    | 354 |
| 7.25.2.24 | <a href="#">nppiSqrt_8u_C1RSfs</a>     | 354 |
| 7.25.2.25 | <a href="#">nppiSqrt_8u_C3IRSfs</a>    | 355 |
| 7.25.2.26 | <a href="#">nppiSqrt_8u_C3RSfs</a>     | 355 |
| 7.26      | <a href="#">Ln</a>                     | 356 |
| 7.26.1    | <a href="#">Detailed Description</a>   | 357 |
| 7.26.2    | <a href="#">Function Documentation</a> | 357 |
| 7.26.2.1  | <a href="#">nppiLn_16s_C1IRSfs</a>     | 357 |
| 7.26.2.2  | <a href="#">nppiLn_16s_C1RSfs</a>      | 358 |
| 7.26.2.3  | <a href="#">nppiLn_16s_C3IRSfs</a>     | 358 |
| 7.26.2.4  | <a href="#">nppiLn_16s_C3RSfs</a>      | 358 |
| 7.26.2.5  | <a href="#">nppiLn_16u_C1IRSfs</a>     | 359 |
| 7.26.2.6  | <a href="#">nppiLn_16u_C1RSfs</a>      | 359 |
| 7.26.2.7  | <a href="#">nppiLn_16u_C3IRSfs</a>     | 359 |
| 7.26.2.8  | <a href="#">nppiLn_16u_C3RSfs</a>      | 360 |
| 7.26.2.9  | <a href="#">nppiLn_32f_C1IR</a>        | 360 |
| 7.26.2.10 | <a href="#">nppiLn_32f_C1R</a>         | 360 |
| 7.26.2.11 | <a href="#">nppiLn_32f_C3IR</a>        | 361 |
| 7.26.2.12 | <a href="#">nppiLn_32f_C3R</a>         | 361 |
| 7.26.2.13 | <a href="#">nppiLn_8u_C1IRSfs</a>      | 361 |
| 7.26.2.14 | <a href="#">nppiLn_8u_C1RSfs</a>       | 362 |
| 7.26.2.15 | <a href="#">nppiLn_8u_C3IRSfs</a>      | 362 |
| 7.26.2.16 | <a href="#">nppiLn_8u_C3RSfs</a>       | 362 |
| 7.27      | <a href="#">Exp</a>                    | 363 |
| 7.27.1    | <a href="#">Detailed Description</a>   | 364 |
| 7.27.2    | <a href="#">Function Documentation</a> | 364 |

|           |  |     |
|-----------|--|-----|
| 7.27.2.1  | <a href="#">nppiExp_16s_C1IRSfs</a>    | 364 |
| 7.27.2.2  | <a href="#">nppiExp_16s_C1RSfs</a>     | 365 |
| 7.27.2.3  | <a href="#">nppiExp_16s_C3IRSfs</a>    | 365 |
| 7.27.2.4  | <a href="#">nppiExp_16s_C3RSfs</a>     | 365 |
| 7.27.2.5  | <a href="#">nppiExp_16u_C1IRSfs</a>    | 366 |
| 7.27.2.6  | <a href="#">nppiExp_16u_C1RSfs</a>     | 366 |
| 7.27.2.7  | <a href="#">nppiExp_16u_C3IRSfs</a>    | 366 |
| 7.27.2.8  | <a href="#">nppiExp_16u_C3RSfs</a>     | 367 |
| 7.27.2.9  | <a href="#">nppiExp_32f_C1IR</a>       | 367 |
| 7.27.2.10 | <a href="#">nppiExp_32f_C1R</a>        | 367 |
| 7.27.2.11 | <a href="#">nppiExp_32f_C3IR</a>       | 368 |
| 7.27.2.12 | <a href="#">nppiExp_32f_C3R</a>        | 368 |
| 7.27.2.13 | <a href="#">nppiExp_8u_C1IRSfs</a>     | 368 |
| 7.27.2.14 | <a href="#">nppiExp_8u_C1RSfs</a>      | 369 |
| 7.27.2.15 | <a href="#">nppiExp_8u_C3IRSfs</a>     | 369 |
| 7.27.2.16 | <a href="#">nppiExp_8u_C3RSfs</a>      | 369 |
| 7.28      | <a href="#">Logical Operations</a>     | 370 |
| 7.29      | <a href="#">AndC</a>                   | 371 |
| 7.29.1    | <a href="#">Detailed Description</a>   | 373 |
| 7.29.2    | <a href="#">Function Documentation</a> | 373 |
| 7.29.2.1  | <a href="#">nppiAndC_16u_AC4IR</a>     | 373 |
| 7.29.2.2  | <a href="#">nppiAndC_16u_AC4R</a>      | 373 |
| 7.29.2.3  | <a href="#">nppiAndC_16u_C1IR</a>      | 373 |
| 7.29.2.4  | <a href="#">nppiAndC_16u_C1R</a>       | 374 |
| 7.29.2.5  | <a href="#">nppiAndC_16u_C3IR</a>      | 374 |
| 7.29.2.6  | <a href="#">nppiAndC_16u_C3R</a>       | 374 |
| 7.29.2.7  | <a href="#">nppiAndC_16u_C4IR</a>      | 375 |
| 7.29.2.8  | <a href="#">nppiAndC_16u_C4R</a>       | 375 |
| 7.29.2.9  | <a href="#">nppiAndC_32s_AC4IR</a>     | 376 |
| 7.29.2.10 | <a href="#">nppiAndC_32s_AC4R</a>      | 376 |
| 7.29.2.11 | <a href="#">nppiAndC_32s_C1IR</a>      | 376 |
| 7.29.2.12 | <a href="#">nppiAndC_32s_C1R</a>       | 377 |
| 7.29.2.13 | <a href="#">nppiAndC_32s_C3IR</a>      | 377 |
| 7.29.2.14 | <a href="#">nppiAndC_32s_C3R</a>       | 377 |
| 7.29.2.15 | <a href="#">nppiAndC_32s_C4IR</a>      | 378 |
| 7.29.2.16 | <a href="#">nppiAndC_32s_C4R</a>       | 378 |

|           |  |     |
|-----------|--|-----|
| 7.29.2.17 | <a href="#">nppiAndC_8u_AC4IR</a>      | 378 |
| 7.29.2.18 | <a href="#">nppiAndC_8u_AC4R</a>       | 379 |
| 7.29.2.19 | <a href="#">nppiAndC_8u_C1IR</a>       | 379 |
| 7.29.2.20 | <a href="#">nppiAndC_8u_C1R</a>        | 379 |
| 7.29.2.21 | <a href="#">nppiAndC_8u_C3IR</a>       | 380 |
| 7.29.2.22 | <a href="#">nppiAndC_8u_C3R</a>        | 380 |
| 7.29.2.23 | <a href="#">nppiAndC_8u_C4IR</a>       | 380 |
| 7.29.2.24 | <a href="#">nppiAndC_8u_C4R</a>        | 381 |
| 7.30      | <a href="#">OrC</a>                    | 382 |
| 7.30.1    | <a href="#">Detailed Description</a>   | 384 |
| 7.30.2    | <a href="#">Function Documentation</a> | 384 |
| 7.30.2.1  | <a href="#">nppiOrC_16u_AC4IR</a>      | 384 |
| 7.30.2.2  | <a href="#">nppiOrC_16u_AC4R</a>       | 384 |
| 7.30.2.3  | <a href="#">nppiOrC_16u_C1IR</a>       | 384 |
| 7.30.2.4  | <a href="#">nppiOrC_16u_C1R</a>        | 385 |
| 7.30.2.5  | <a href="#">nppiOrC_16u_C3IR</a>       | 385 |
| 7.30.2.6  | <a href="#">nppiOrC_16u_C3R</a>        | 385 |
| 7.30.2.7  | <a href="#">nppiOrC_16u_C4IR</a>       | 386 |
| 7.30.2.8  | <a href="#">nppiOrC_16u_C4R</a>        | 386 |
| 7.30.2.9  | <a href="#">nppiOrC_32s_AC4IR</a>      | 387 |
| 7.30.2.10 | <a href="#">nppiOrC_32s_AC4R</a>       | 387 |
| 7.30.2.11 | <a href="#">nppiOrC_32s_C1IR</a>       | 387 |
| 7.30.2.12 | <a href="#">nppiOrC_32s_C1R</a>        | 388 |
| 7.30.2.13 | <a href="#">nppiOrC_32s_C3IR</a>       | 388 |
| 7.30.2.14 | <a href="#">nppiOrC_32s_C3R</a>        | 388 |
| 7.30.2.15 | <a href="#">nppiOrC_32s_C4IR</a>       | 389 |
| 7.30.2.16 | <a href="#">nppiOrC_32s_C4R</a>        | 389 |
| 7.30.2.17 | <a href="#">nppiOrC_8u_AC4IR</a>       | 389 |
| 7.30.2.18 | <a href="#">nppiOrC_8u_AC4R</a>        | 390 |
| 7.30.2.19 | <a href="#">nppiOrC_8u_C1IR</a>        | 390 |
| 7.30.2.20 | <a href="#">nppiOrC_8u_C1R</a>         | 390 |
| 7.30.2.21 | <a href="#">nppiOrC_8u_C3IR</a>        | 391 |
| 7.30.2.22 | <a href="#">nppiOrC_8u_C3R</a>         | 391 |
| 7.30.2.23 | <a href="#">nppiOrC_8u_C4IR</a>        | 391 |
| 7.30.2.24 | <a href="#">nppiOrC_8u_C4R</a>         | 392 |
| 7.31      | <a href="#">XorC</a>                   | 393 |

|  |     |
|--|-----|
| 7.31.1 Detailed Description . . . . .    | 395 |
| 7.31.2 Function Documentation . . . . .  | 395 |
| 7.31.2.1 nppiXorC_16u_AC4IR . . . . .    | 395 |
| 7.31.2.2 nppiXorC_16u_AC4R . . . . .     | 395 |
| 7.31.2.3 nppiXorC_16u_C1IR . . . . .     | 395 |
| 7.31.2.4 nppiXorC_16u_C1R . . . . .      | 396 |
| 7.31.2.5 nppiXorC_16u_C3IR . . . . .     | 396 |
| 7.31.2.6 nppiXorC_16u_C3R . . . . .      | 396 |
| 7.31.2.7 nppiXorC_16u_C4IR . . . . .     | 397 |
| 7.31.2.8 nppiXorC_16u_C4R . . . . .      | 397 |
| 7.31.2.9 nppiXorC_32s_AC4IR . . . . .    | 398 |
| 7.31.2.10 nppiXorC_32s_AC4R . . . . .    | 398 |
| 7.31.2.11 nppiXorC_32s_C1IR . . . . .    | 398 |
| 7.31.2.12 nppiXorC_32s_C1R . . . . .     | 399 |
| 7.31.2.13 nppiXorC_32s_C3IR . . . . .    | 399 |
| 7.31.2.14 nppiXorC_32s_C3R . . . . .     | 399 |
| 7.31.2.15 nppiXorC_32s_C4IR . . . . .    | 400 |
| 7.31.2.16 nppiXorC_32s_C4R . . . . .     | 400 |
| 7.31.2.17 nppiXorC_8u_AC4IR . . . . .    | 400 |
| 7.31.2.18 nppiXorC_8u_AC4R . . . . .     | 401 |
| 7.31.2.19 nppiXorC_8u_C1IR . . . . .     | 401 |
| 7.31.2.20 nppiXorC_8u_C1R . . . . .      | 401 |
| 7.31.2.21 nppiXorC_8u_C3IR . . . . .     | 402 |
| 7.31.2.22 nppiXorC_8u_C3R . . . . .      | 402 |
| 7.31.2.23 nppiXorC_8u_C4IR . . . . .     | 402 |
| 7.31.2.24 nppiXorC_8u_C4R . . . . .      | 403 |
| 7.32 RShiftC . . . . .                   | 404 |
| 7.32.1 Detailed Description . . . . .    | 407 |
| 7.32.2 Function Documentation . . . . .  | 407 |
| 7.32.2.1 nppiRShiftC_16s_AC4IR . . . . . | 407 |
| 7.32.2.2 nppiRShiftC_16s_AC4R . . . . .  | 407 |
| 7.32.2.3 nppiRShiftC_16s_C1IR . . . . .  | 408 |
| 7.32.2.4 nppiRShiftC_16s_C1R . . . . .   | 408 |
| 7.32.2.5 nppiRShiftC_16s_C3IR . . . . .  | 408 |
| 7.32.2.6 nppiRShiftC_16s_C3R . . . . .   | 409 |
| 7.32.2.7 nppiRShiftC_16s_C4IR . . . . .  | 409 |



|           |                                    |     |
|-----------|------------------------------------|-----|
| 7.32.2.8  | <code>nppiRShiftC_16s_C4R</code>   | 409 |
| 7.32.2.9  | <code>nppiRShiftC_16u_AC4IR</code> | 410 |
| 7.32.2.10 | <code>nppiRShiftC_16u_AC4R</code>  | 410 |
| 7.32.2.11 | <code>nppiRShiftC_16u_C1IR</code>  | 411 |
| 7.32.2.12 | <code>nppiRShiftC_16u_C1R</code>   | 411 |
| 7.32.2.13 | <code>nppiRShiftC_16u_C3IR</code>  | 411 |
| 7.32.2.14 | <code>nppiRShiftC_16u_C3R</code>   | 412 |
| 7.32.2.15 | <code>nppiRShiftC_16u_C4IR</code>  | 412 |
| 7.32.2.16 | <code>nppiRShiftC_16u_C4R</code>   | 412 |
| 7.32.2.17 | <code>nppiRShiftC_32s_AC4IR</code> | 413 |
| 7.32.2.18 | <code>nppiRShiftC_32s_AC4R</code>  | 413 |
| 7.32.2.19 | <code>nppiRShiftC_32s_C1IR</code>  | 413 |
| 7.32.2.20 | <code>nppiRShiftC_32s_C1R</code>   | 414 |
| 7.32.2.21 | <code>nppiRShiftC_32s_C3IR</code>  | 414 |
| 7.32.2.22 | <code>nppiRShiftC_32s_C3R</code>   | 414 |
| 7.32.2.23 | <code>nppiRShiftC_32s_C4IR</code>  | 415 |
| 7.32.2.24 | <code>nppiRShiftC_32s_C4R</code>   | 415 |
| 7.32.2.25 | <code>nppiRShiftC_8s_AC4IR</code>  | 415 |
| 7.32.2.26 | <code>nppiRShiftC_8s_AC4R</code>   | 416 |
| 7.32.2.27 | <code>nppiRShiftC_8s_C1IR</code>   | 416 |
| 7.32.2.28 | <code>nppiRShiftC_8s_C1R</code>    | 416 |
| 7.32.2.29 | <code>nppiRShiftC_8s_C3IR</code>   | 417 |
| 7.32.2.30 | <code>nppiRShiftC_8s_C3R</code>    | 417 |
| 7.32.2.31 | <code>nppiRShiftC_8s_C4IR</code>   | 417 |
| 7.32.2.32 | <code>nppiRShiftC_8s_C4R</code>    | 418 |
| 7.32.2.33 | <code>nppiRShiftC_8u_AC4IR</code>  | 418 |
| 7.32.2.34 | <code>nppiRShiftC_8u_AC4R</code>   | 418 |
| 7.32.2.35 | <code>nppiRShiftC_8u_C1IR</code>   | 419 |
| 7.32.2.36 | <code>nppiRShiftC_8u_C1R</code>    | 419 |
| 7.32.2.37 | <code>nppiRShiftC_8u_C3IR</code>   | 419 |
| 7.32.2.38 | <code>nppiRShiftC_8u_C3R</code>    | 420 |
| 7.32.2.39 | <code>nppiRShiftC_8u_C4IR</code>   | 420 |
| 7.32.2.40 | <code>nppiRShiftC_8u_C4R</code>    | 420 |
| 7.33      | <code>LShiftC</code>               | 421 |
| 7.33.1    | Detailed Description               | 423 |
| 7.33.2    | Function Documentation             | 423 |

|           |                                    |     |
|-----------|------------------------------------|-----|
| 7.33.2.1  | <code>nppiLShiftC_16u_AC4IR</code> | 423 |
| 7.33.2.2  | <code>nppiLShiftC_16u_AC4R</code>  | 423 |
| 7.33.2.3  | <code>nppiLShiftC_16u_C1IR</code>  | 423 |
| 7.33.2.4  | <code>nppiLShiftC_16u_C1R</code>   | 424 |
| 7.33.2.5  | <code>nppiLShiftC_16u_C3IR</code>  | 424 |
| 7.33.2.6  | <code>nppiLShiftC_16u_C3R</code>   | 424 |
| 7.33.2.7  | <code>nppiLShiftC_16u_C4IR</code>  | 425 |
| 7.33.2.8  | <code>nppiLShiftC_16u_C4R</code>   | 425 |
| 7.33.2.9  | <code>nppiLShiftC_32s_AC4IR</code> | 426 |
| 7.33.2.10 | <code>nppiLShiftC_32s_AC4R</code>  | 426 |
| 7.33.2.11 | <code>nppiLShiftC_32s_C1IR</code>  | 426 |
| 7.33.2.12 | <code>nppiLShiftC_32s_C1R</code>   | 427 |
| 7.33.2.13 | <code>nppiLShiftC_32s_C3IR</code>  | 427 |
| 7.33.2.14 | <code>nppiLShiftC_32s_C3R</code>   | 427 |
| 7.33.2.15 | <code>nppiLShiftC_32s_C4IR</code>  | 428 |
| 7.33.2.16 | <code>nppiLShiftC_32s_C4R</code>   | 428 |
| 7.33.2.17 | <code>nppiLShiftC_8u_AC4IR</code>  | 428 |
| 7.33.2.18 | <code>nppiLShiftC_8u_AC4R</code>   | 429 |
| 7.33.2.19 | <code>nppiLShiftC_8u_C1IR</code>   | 429 |
| 7.33.2.20 | <code>nppiLShiftC_8u_C1R</code>    | 429 |
| 7.33.2.21 | <code>nppiLShiftC_8u_C3IR</code>   | 430 |
| 7.33.2.22 | <code>nppiLShiftC_8u_C3R</code>    | 430 |
| 7.33.2.23 | <code>nppiLShiftC_8u_C4IR</code>   | 430 |
| 7.33.2.24 | <code>nppiLShiftC_8u_C4R</code>    | 431 |
| 7.34      | <b>And</b>                         | 432 |
| 7.34.1    | Detailed Description               | 434 |
| 7.34.2    | Function Documentation             | 434 |
| 7.34.2.1  | <code>nppiAnd_16u_AC4IR</code>     | 434 |
| 7.34.2.2  | <code>nppiAnd_16u_AC4R</code>      | 434 |
| 7.34.2.3  | <code>nppiAnd_16u_C1IR</code>      | 434 |
| 7.34.2.4  | <code>nppiAnd_16u_C1R</code>       | 435 |
| 7.34.2.5  | <code>nppiAnd_16u_C3IR</code>      | 435 |
| 7.34.2.6  | <code>nppiAnd_16u_C3R</code>       | 436 |
| 7.34.2.7  | <code>nppiAnd_16u_C4IR</code>      | 436 |
| 7.34.2.8  | <code>nppiAnd_16u_C4R</code>       | 436 |
| 7.34.2.9  | <code>nppiAnd_32s_AC4IR</code>     | 437 |

|           |                               |     |
|-----------|-------------------------------|-----|
| 7.34.2.10 | <code>nppiAnd_32s_AC4R</code> | 437 |
| 7.34.2.11 | <code>nppiAnd_32s_C1IR</code> | 438 |
| 7.34.2.12 | <code>nppiAnd_32s_C1R</code>  | 438 |
| 7.34.2.13 | <code>nppiAnd_32s_C3IR</code> | 438 |
| 7.34.2.14 | <code>nppiAnd_32s_C3R</code>  | 439 |
| 7.34.2.15 | <code>nppiAnd_32s_C4IR</code> | 439 |
| 7.34.2.16 | <code>nppiAnd_32s_C4R</code>  | 439 |
| 7.34.2.17 | <code>nppiAnd_8u_AC4IR</code> | 440 |
| 7.34.2.18 | <code>nppiAnd_8u_AC4R</code>  | 440 |
| 7.34.2.19 | <code>nppiAnd_8u_C1IR</code>  | 441 |
| 7.34.2.20 | <code>nppiAnd_8u_C1R</code>   | 441 |
| 7.34.2.21 | <code>nppiAnd_8u_C3IR</code>  | 441 |
| 7.34.2.22 | <code>nppiAnd_8u_C3R</code>   | 442 |
| 7.34.2.23 | <code>nppiAnd_8u_C4IR</code>  | 442 |
| 7.34.2.24 | <code>nppiAnd_8u_C4R</code>   | 442 |
| 7.35      | Or                            | 444 |
| 7.35.1    | Detailed Description          | 446 |
| 7.35.2    | Function Documentation        | 446 |
| 7.35.2.1  | <code>nppiOr_16u_AC4IR</code> | 446 |
| 7.35.2.2  | <code>nppiOr_16u_AC4R</code>  | 446 |
| 7.35.2.3  | <code>nppiOr_16u_C1IR</code>  | 446 |
| 7.35.2.4  | <code>nppiOr_16u_C1R</code>   | 447 |
| 7.35.2.5  | <code>nppiOr_16u_C3IR</code>  | 447 |
| 7.35.2.6  | <code>nppiOr_16u_C3R</code>   | 448 |
| 7.35.2.7  | <code>nppiOr_16u_C4IR</code>  | 448 |
| 7.35.2.8  | <code>nppiOr_16u_C4R</code>   | 448 |
| 7.35.2.9  | <code>nppiOr_32s_AC4IR</code> | 449 |
| 7.35.2.10 | <code>nppiOr_32s_AC4R</code>  | 449 |
| 7.35.2.11 | <code>nppiOr_32s_C1IR</code>  | 450 |
| 7.35.2.12 | <code>nppiOr_32s_C1R</code>   | 450 |
| 7.35.2.13 | <code>nppiOr_32s_C3IR</code>  | 450 |
| 7.35.2.14 | <code>nppiOr_32s_C3R</code>   | 451 |
| 7.35.2.15 | <code>nppiOr_32s_C4IR</code>  | 451 |
| 7.35.2.16 | <code>nppiOr_32s_C4R</code>   | 451 |
| 7.35.2.17 | <code>nppiOr_8u_AC4IR</code>  | 452 |
| 7.35.2.18 | <code>nppiOr_8u_AC4R</code>   | 452 |

|           |  |     |
|-----------|--|-----|
| 7.35.2.19 | <a href="#">nppiOr_8u_C1IR</a>         | 453 |
| 7.35.2.20 | <a href="#">nppiOr_8u_C1R</a>          | 453 |
| 7.35.2.21 | <a href="#">nppiOr_8u_C3IR</a>         | 453 |
| 7.35.2.22 | <a href="#">nppiOr_8u_C3R</a>          | 454 |
| 7.35.2.23 | <a href="#">nppiOr_8u_C4IR</a>         | 454 |
| 7.35.2.24 | <a href="#">nppiOr_8u_C4R</a>          | 454 |
| 7.36      | <a href="#">Xor</a>                    | 456 |
| 7.36.1    | <a href="#">Detailed Description</a>   | 458 |
| 7.36.2    | <a href="#">Function Documentation</a> | 458 |
| 7.36.2.1  | <a href="#">nppiXor_16u_AC4IR</a>      | 458 |
| 7.36.2.2  | <a href="#">nppiXor_16u_AC4R</a>       | 458 |
| 7.36.2.3  | <a href="#">nppiXor_16u_C1IR</a>       | 458 |
| 7.36.2.4  | <a href="#">nppiXor_16u_C1R</a>        | 459 |
| 7.36.2.5  | <a href="#">nppiXor_16u_C3IR</a>       | 459 |
| 7.36.2.6  | <a href="#">nppiXor_16u_C3R</a>        | 460 |
| 7.36.2.7  | <a href="#">nppiXor_16u_C4IR</a>       | 460 |
| 7.36.2.8  | <a href="#">nppiXor_16u_C4R</a>        | 460 |
| 7.36.2.9  | <a href="#">nppiXor_32s_AC4IR</a>      | 461 |
| 7.36.2.10 | <a href="#">nppiXor_32s_AC4R</a>       | 461 |
| 7.36.2.11 | <a href="#">nppiXor_32s_C1IR</a>       | 462 |
| 7.36.2.12 | <a href="#">nppiXor_32s_C1R</a>        | 462 |
| 7.36.2.13 | <a href="#">nppiXor_32s_C3IR</a>       | 462 |
| 7.36.2.14 | <a href="#">nppiXor_32s_C3R</a>        | 463 |
| 7.36.2.15 | <a href="#">nppiXor_32s_C4IR</a>       | 463 |
| 7.36.2.16 | <a href="#">nppiXor_32s_C4R</a>        | 463 |
| 7.36.2.17 | <a href="#">nppiXor_8u_AC4IR</a>       | 464 |
| 7.36.2.18 | <a href="#">nppiXor_8u_AC4R</a>        | 464 |
| 7.36.2.19 | <a href="#">nppiXor_8u_C1IR</a>        | 465 |
| 7.36.2.20 | <a href="#">nppiXor_8u_C1R</a>         | 465 |
| 7.36.2.21 | <a href="#">nppiXor_8u_C3IR</a>        | 465 |
| 7.36.2.22 | <a href="#">nppiXor_8u_C3R</a>         | 466 |
| 7.36.2.23 | <a href="#">nppiXor_8u_C4IR</a>        | 466 |
| 7.36.2.24 | <a href="#">nppiXor_8u_C4R</a>         | 466 |
| 7.37      | <a href="#">Not</a>                    | 468 |
| 7.37.1    | <a href="#">Detailed Description</a>   | 468 |
| 7.37.2    | <a href="#">Function Documentation</a> | 468 |

|           |  |     |
|-----------|--|-----|
| 7.37.2.1  | <a href="#">nppiNot_8u_AC4IR</a>           | 468 |
| 7.37.2.2  | <a href="#">nppiNot_8u_AC4R</a>            | 469 |
| 7.37.2.3  | <a href="#">nppiNot_8u_C1IR</a>            | 469 |
| 7.37.2.4  | <a href="#">nppiNot_8u_C1R</a>             | 469 |
| 7.37.2.5  | <a href="#">nppiNot_8u_C3IR</a>            | 470 |
| 7.37.2.6  | <a href="#">nppiNot_8u_C3R</a>             | 470 |
| 7.37.2.7  | <a href="#">nppiNot_8u_C4IR</a>            | 470 |
| 7.37.2.8  | <a href="#">nppiNot_8u_C4R</a>             | 471 |
| 7.38      | <a href="#">Alpha Composition</a>          | 472 |
| 7.39      | <a href="#">AlphaCompC</a>                 | 473 |
| 7.39.1    | <a href="#">Detailed Description</a>       | 474 |
| 7.39.2    | <a href="#">Function Documentation</a>     | 474 |
| 7.39.2.1  | <a href="#">nppiAlphaCompC_16s_C1R</a>     | 474 |
| 7.39.2.2  | <a href="#">nppiAlphaCompC_16u_AC4R</a>    | 475 |
| 7.39.2.3  | <a href="#">nppiAlphaCompC_16u_C1R</a>     | 475 |
| 7.39.2.4  | <a href="#">nppiAlphaCompC_16u_C3R</a>     | 476 |
| 7.39.2.5  | <a href="#">nppiAlphaCompC_16u_C4R</a>     | 476 |
| 7.39.2.6  | <a href="#">nppiAlphaCompC_32f_C1R</a>     | 477 |
| 7.39.2.7  | <a href="#">nppiAlphaCompC_32s_C1R</a>     | 477 |
| 7.39.2.8  | <a href="#">nppiAlphaCompC_32u_C1R</a>     | 478 |
| 7.39.2.9  | <a href="#">nppiAlphaCompC_8s_C1R</a>      | 478 |
| 7.39.2.10 | <a href="#">nppiAlphaCompC_8u_AC4R</a>     | 479 |
| 7.39.2.11 | <a href="#">nppiAlphaCompC_8u_C1R</a>      | 479 |
| 7.39.2.12 | <a href="#">nppiAlphaCompC_8u_C3R</a>      | 480 |
| 7.39.2.13 | <a href="#">nppiAlphaCompC_8u_C4R</a>      | 480 |
| 7.40      | <a href="#">AlphaPremulC</a>               | 481 |
| 7.40.1    | <a href="#">Detailed Description</a>       | 482 |
| 7.40.2    | <a href="#">Function Documentation</a>     | 482 |
| 7.40.2.1  | <a href="#">nppiAlphaPremulC_16u_AC4IR</a> | 482 |
| 7.40.2.2  | <a href="#">nppiAlphaPremulC_16u_AC4R</a>  | 482 |
| 7.40.2.3  | <a href="#">nppiAlphaPremulC_16u_C1IR</a>  | 483 |
| 7.40.2.4  | <a href="#">nppiAlphaPremulC_16u_C1R</a>   | 483 |
| 7.40.2.5  | <a href="#">nppiAlphaPremulC_16u_C3IR</a>  | 484 |
| 7.40.2.6  | <a href="#">nppiAlphaPremulC_16u_C3R</a>   | 484 |
| 7.40.2.7  | <a href="#">nppiAlphaPremulC_16u_C4IR</a>  | 484 |
| 7.40.2.8  | <a href="#">nppiAlphaPremulC_16u_C4R</a>   | 485 |

|           |   |     |
|-----------|---|-----|
| 7.40.2.9  | <a href="#">nppiAlphaPremulC_8u_AC4IR</a>           | 485 |
| 7.40.2.10 | <a href="#">nppiAlphaPremulC_8u_AC4R</a>            | 485 |
| 7.40.2.11 | <a href="#">nppiAlphaPremulC_8u_C1IR</a>            | 486 |
| 7.40.2.12 | <a href="#">nppiAlphaPremulC_8u_C1R</a>             | 486 |
| 7.40.2.13 | <a href="#">nppiAlphaPremulC_8u_C3IR</a>            | 486 |
| 7.40.2.14 | <a href="#">nppiAlphaPremulC_8u_C3R</a>             | 487 |
| 7.40.2.15 | <a href="#">nppiAlphaPremulC_8u_C4IR</a>            | 487 |
| 7.40.2.16 | <a href="#">nppiAlphaPremulC_8u_C4R</a>             | 487 |
| 7.41      | <a href="#">AlphaComp</a>                           | 488 |
| 7.41.1    | <a href="#">Detailed Description</a>                | 489 |
| 7.41.2    | <a href="#">Function Documentation</a>              | 489 |
| 7.41.2.1  | <a href="#">nppiAlphaComp_16s_AC1R</a>              | 489 |
| 7.41.2.2  | <a href="#">nppiAlphaComp_16u_AC1R</a>              | 489 |
| 7.41.2.3  | <a href="#">nppiAlphaComp_16u_AC4R</a>              | 490 |
| 7.41.2.4  | <a href="#">nppiAlphaComp_32f_AC1R</a>              | 490 |
| 7.41.2.5  | <a href="#">nppiAlphaComp_32f_AC4R</a>              | 491 |
| 7.41.2.6  | <a href="#">nppiAlphaComp_32s_AC1R</a>              | 491 |
| 7.41.2.7  | <a href="#">nppiAlphaComp_32s_AC4R</a>              | 492 |
| 7.41.2.8  | <a href="#">nppiAlphaComp_32u_AC1R</a>              | 492 |
| 7.41.2.9  | <a href="#">nppiAlphaComp_32u_AC4R</a>              | 493 |
| 7.41.2.10 | <a href="#">nppiAlphaComp_8s_AC1R</a>               | 493 |
| 7.41.2.11 | <a href="#">nppiAlphaComp_8u_AC1R</a>               | 493 |
| 7.41.2.12 | <a href="#">nppiAlphaComp_8u_AC4R</a>               | 494 |
| 7.42      | <a href="#">AlphaPremul</a>                         | 495 |
| 7.42.1    | <a href="#">Detailed Description</a>                | 495 |
| 7.42.2    | <a href="#">Function Documentation</a>              | 495 |
| 7.42.2.1  | <a href="#">nppiAlphaPremul_16u_AC4IR</a>           | 495 |
| 7.42.2.2  | <a href="#">nppiAlphaPremul_16u_AC4R</a>            | 496 |
| 7.42.2.3  | <a href="#">nppiAlphaPremul_8u_AC4IR</a>            | 496 |
| 7.42.2.4  | <a href="#">nppiAlphaPremul_8u_AC4R</a>             | 496 |
| 7.43      | <a href="#">Color and Sampling Conversion</a>       | 497 |
| 7.43.1    | <a href="#">Detailed Description</a>                | 497 |
| 7.44      | <a href="#">Color Model Conversion</a>              | 498 |
| 7.44.1    | <a href="#">Detailed Description</a>                | 524 |
| 7.44.2    | <a href="#">Function Documentation</a>              | 525 |
| 7.44.2.1  | <a href="#">nppiBGRToCbYCr422_709HDTV_8u_AC4C2R</a> | 525 |

|           |  |     |
|-----------|--|-----|
| 7.44.2.2  | <a href="#">nppiBGRTToCbYCr422_709HDTV_8u_C3C2R</a>  | 525 |
| 7.44.2.3  | <a href="#">nppiBGRTToCbYCr422_8u_AC4C2R</a>         | 525 |
| 7.44.2.4  | <a href="#">nppiBGRTToHLS_8u_AC4P4R</a>              | 526 |
| 7.44.2.5  | <a href="#">nppiBGRTToHLS_8u_AC4R</a>                | 526 |
| 7.44.2.6  | <a href="#">nppiBGRTToHLS_8u_AP4C4R</a>              | 527 |
| 7.44.2.7  | <a href="#">nppiBGRTToHLS_8u_AP4R</a>                | 527 |
| 7.44.2.8  | <a href="#">nppiBGRTToHLS_8u_C3P3R</a>               | 527 |
| 7.44.2.9  | <a href="#">nppiBGRTToHLS_8u_P3C3R</a>               | 528 |
| 7.44.2.10 | <a href="#">nppiBGRTToHLS_8u_P3R</a>                 | 528 |
| 7.44.2.11 | <a href="#">nppiBGRTToLab_8u_C3R</a>                 | 528 |
| 7.44.2.12 | <a href="#">nppiBGRTToYCbCr411_8u_AC4P3R</a>         | 529 |
| 7.44.2.13 | <a href="#">nppiBGRTToYCbCr411_8u_C3P3R</a>          | 529 |
| 7.44.2.14 | <a href="#">nppiBGRTToYCbCr420_709CSC_8u_AC4P3R</a>  | 529 |
| 7.44.2.15 | <a href="#">nppiBGRTToYCbCr420_709CSC_8u_C3P3R</a>   | 530 |
| 7.44.2.16 | <a href="#">nppiBGRTToYCbCr420_709HDTV_8u_AC4P3R</a> | 530 |
| 7.44.2.17 | <a href="#">nppiBGRTToYCbCr420_8u_AC4P3R</a>         | 531 |
| 7.44.2.18 | <a href="#">nppiBGRTToYCbCr420_8u_C3P3R</a>          | 531 |
| 7.44.2.19 | <a href="#">nppiBGRTToYCbCr422_8u_AC4C2R</a>         | 531 |
| 7.44.2.20 | <a href="#">nppiBGRTToYCbCr422_8u_AC4P3R</a>         | 532 |
| 7.44.2.21 | <a href="#">nppiBGRTToYCbCr422_8u_C3C2R</a>          | 532 |
| 7.44.2.22 | <a href="#">nppiBGRTToYCbCr422_8u_C3P3R</a>          | 533 |
| 7.44.2.23 | <a href="#">nppiBGRTToYCbCr_8u_AC4P3R</a>            | 533 |
| 7.44.2.24 | <a href="#">nppiBGRTToYCbCr_8u_AC4P4R</a>            | 533 |
| 7.44.2.25 | <a href="#">nppiBGRTToYCbCr_8u_C3P3R</a>             | 534 |
| 7.44.2.26 | <a href="#">nppiBGRTToYCrCb420_709CSC_8u_AC4P3R</a>  | 534 |
| 7.44.2.27 | <a href="#">nppiBGRTToYCrCb420_709CSC_8u_C3P3R</a>   | 535 |
| 7.44.2.28 | <a href="#">nppiBGRTToYCrCb420_8u_AC4P3R</a>         | 535 |
| 7.44.2.29 | <a href="#">nppiBGRTToYCrCb420_8u_C3P3R</a>          | 535 |
| 7.44.2.30 | <a href="#">nppiBGRTToYUV420_8u_AC4P3R</a>           | 536 |
| 7.44.2.31 | <a href="#">nppiBGRTToYUV_8u_AC4P4R</a>              | 536 |
| 7.44.2.32 | <a href="#">nppiBGRTToYUV_8u_AC4R</a>                | 537 |
| 7.44.2.33 | <a href="#">nppiBGRTToYUV_8u_C3P3R</a>               | 537 |
| 7.44.2.34 | <a href="#">nppiBGRTToYUV_8u_C3R</a>                 | 537 |
| 7.44.2.35 | <a href="#">nppiBGRTToYUV_8u_P3R</a>                 | 538 |
| 7.44.2.36 | <a href="#">nppiCbYCr422ToBGR_709HDTV_8u_C2C3R</a>   | 538 |
| 7.44.2.37 | <a href="#">nppiCbYCr422ToBGR_709HDTV_8u_C2C4R</a>   | 538 |

|   |     |
|---|-----|
| 7.44.2.38 nppiCbYCr422ToBGR_8u_C2C4R . . . . .      | 539 |
| 7.44.2.39 nppiCbYCr422ToRGB_8u_C2C3R . . . . .      | 539 |
| 7.44.2.40 nppiColorToGray_16s_AC4C1R . . . . .      | 540 |
| 7.44.2.41 nppiColorToGray_16s_C3C1R . . . . .       | 540 |
| 7.44.2.42 nppiColorToGray_16u_AC4C1R . . . . .      | 540 |
| 7.44.2.43 nppiColorToGray_16u_C3C1R . . . . .       | 541 |
| 7.44.2.44 nppiColorToGray_32f_AC4C1R . . . . .      | 541 |
| 7.44.2.45 nppiColorToGray_32f_C3C1R . . . . .       | 541 |
| 7.44.2.46 nppiColorToGray_8u_AC4C1R . . . . .       | 542 |
| 7.44.2.47 nppiColorToGray_8u_C3C1R . . . . .        | 542 |
| 7.44.2.48 nppiHLSToBGR_8u_AC4P4R . . . . .          | 543 |
| 7.44.2.49 nppiHLSToBGR_8u_AC4R . . . . .            | 543 |
| 7.44.2.50 nppiHLSToBGR_8u_AP4C4R . . . . .          | 543 |
| 7.44.2.51 nppiHLSToBGR_8u_AP4R . . . . .            | 544 |
| 7.44.2.52 nppiHLSToBGR_8u_C3P3R . . . . .           | 544 |
| 7.44.2.53 nppiHLSToBGR_8u_P3C3R . . . . .           | 544 |
| 7.44.2.54 nppiHLSToBGR_8u_P3R . . . . .             | 545 |
| 7.44.2.55 nppiHLSToRGB_8u_AC4R . . . . .            | 545 |
| 7.44.2.56 nppiHLSToRGB_8u_C3R . . . . .             | 545 |
| 7.44.2.57 nppiHSVToRGB_8u_AC4R . . . . .            | 546 |
| 7.44.2.58 nppiHSVToRGB_8u_C3R . . . . .             | 546 |
| 7.44.2.59 nppiLabToBGR_8u_C3R . . . . .             | 546 |
| 7.44.2.60 nppiLUVToRGB_8u_AC4R . . . . .            | 547 |
| 7.44.2.61 nppiLUVToRGB_8u_C3R . . . . .             | 547 |
| 7.44.2.62 nppiRGBToCbYCr422_8u_C3C2R . . . . .      | 547 |
| 7.44.2.63 nppiRGBToCbYCr422Gamma_8u_C3C2R . . . . . | 548 |
| 7.44.2.64 nppiRGBToGray_16s_AC4C1R . . . . .        | 548 |
| 7.44.2.65 nppiRGBToGray_16s_C3C1R . . . . .         | 548 |
| 7.44.2.66 nppiRGBToGray_16u_AC4C1R . . . . .        | 549 |
| 7.44.2.67 nppiRGBToGray_16u_C3C1R . . . . .         | 549 |
| 7.44.2.68 nppiRGBToGray_32f_AC4C1R . . . . .        | 549 |
| 7.44.2.69 nppiRGBToGray_32f_C3C1R . . . . .         | 550 |
| 7.44.2.70 nppiRGBToGray_8u_AC4C1R . . . . .         | 550 |
| 7.44.2.71 nppiRGBToGray_8u_C3C1R . . . . .          | 550 |
| 7.44.2.72 nppiRGBToHLS_8u_AC4R . . . . .            | 551 |
| 7.44.2.73 nppiRGBToHLS_8u_C3R . . . . .             | 551 |



|            |                                    |     |
|------------|------------------------------------|-----|
| 7.44.2.74  | npplRGBToHSV_8u_AC4R               | 551 |
| 7.44.2.75  | npplRGBToHSV_8u_C3R                | 552 |
| 7.44.2.76  | npplRGBToLUV_8u_AC4R               | 552 |
| 7.44.2.77  | npplRGBToLUV_8u_C3R                | 552 |
| 7.44.2.78  | npplRGBToXYZ_8u_AC4R               | 553 |
| 7.44.2.79  | npplRGBToXYZ_8u_C3R                | 553 |
| 7.44.2.80  | npplRGBToYCbCr420_8u_C3P3R         | 553 |
| 7.44.2.81  | npplRGBToYCbCr422_8u_C3C2R         | 554 |
| 7.44.2.82  | npplRGBToYCbCr422_8u_C3P3R         | 554 |
| 7.44.2.83  | npplRGBToYCbCr422_8u_P3C2R         | 554 |
| 7.44.2.84  | npplRGBToYCbCr_8u_AC4P3R           | 555 |
| 7.44.2.85  | npplRGBToYCbCr_8u_AC4R             | 555 |
| 7.44.2.86  | npplRGBToYCbCr_8u_C3P3R            | 555 |
| 7.44.2.87  | npplRGBToYCbCr_8u_C3R              | 556 |
| 7.44.2.88  | npplRGBToYCbCr_8u_P3R              | 556 |
| 7.44.2.89  | npplRGBToYCC_8u_AC4R               | 557 |
| 7.44.2.90  | npplRGBToYCC_8u_C3R                | 557 |
| 7.44.2.91  | npplRGBToYCrCb420_8u_AC4P3R        | 557 |
| 7.44.2.92  | npplRGBToYCrCb422_8u_C3C2R         | 558 |
| 7.44.2.93  | npplRGBToYCrCb422_8u_P3C2R         | 558 |
| 7.44.2.94  | npplRGBToYUV420_8u_C3P3R           | 558 |
| 7.44.2.95  | npplRGBToYUV420_8u_P3R             | 559 |
| 7.44.2.96  | npplRGBToYUV422_8u_C3C2R           | 559 |
| 7.44.2.97  | npplRGBToYUV422_8u_C3P3R           | 559 |
| 7.44.2.98  | npplRGBToYUV422_8u_P3R             | 560 |
| 7.44.2.99  | npplRGBToYUV_8u_AC4P4R             | 560 |
| 7.44.2.100 | npplRGBToYUV_8u_AC4R               | 560 |
| 7.44.2.101 | npplRGBToYUV_8u_C3P3R              | 561 |
| 7.44.2.102 | npplRGBToYUV_8u_C3R                | 561 |
| 7.44.2.103 | npplRGBToYUV_8u_P3R                | 561 |
| 7.44.2.104 | npplXYZToRGB_8u_AC4R               | 562 |
| 7.44.2.105 | npplXYZToRGB_8u_C3R                | 562 |
| 7.44.2.106 | npplYCbCr411ToBGR_8u_P3C3R         | 563 |
| 7.44.2.107 | npplYCbCr411ToBGR_8u_P3C4R         | 563 |
| 7.44.2.108 | npplYCbCr420ToBGR_709CSC_8u_P3C3R  | 563 |
| 7.44.2.109 | npplYCbCr420ToBGR_709HDTV_8u_P3C4R | 564 |

|  |     |
|--|-----|
| 7.44.2.110nppiYCbCr420ToBGR_8u_P3C3R . . . . .     | 564 |
| 7.44.2.111nppiYCbCr420ToBGR_8u_P3C4R . . . . .     | 564 |
| 7.44.2.112nppiYCbCr420ToRGB_8u_P3C3R . . . . .     | 565 |
| 7.44.2.113nppiYCbCr422ToBGR_8u_C2C3R . . . . .     | 565 |
| 7.44.2.114nppiYCbCr422ToBGR_8u_C2C4R . . . . .     | 565 |
| 7.44.2.115nppiYCbCr422ToBGR_8u_P3C3R . . . . .     | 566 |
| 7.44.2.116nppiYCbCr422ToRGB_8u_C2C3R . . . . .     | 566 |
| 7.44.2.117nppiYCbCr422ToRGB_8u_C2P3R . . . . .     | 567 |
| 7.44.2.118nppiYCbCr422ToRGB_8u_P3C3R . . . . .     | 567 |
| 7.44.2.119nppiYCbCrToBGR_709CSC_8u_P3C3R . . . . . | 567 |
| 7.44.2.120nppiYCbCrToBGR_709CSC_8u_P3C4R . . . . . | 568 |
| 7.44.2.121nppiYCbCrToBGR_8u_P3C3R . . . . .        | 568 |
| 7.44.2.122nppiYCbCrToBGR_8u_P3C4R . . . . .        | 568 |
| 7.44.2.123nppiYCbCrToRGB_8u_AC4R . . . . .         | 569 |
| 7.44.2.124nppiYCbCrToRGB_8u_C3R . . . . .          | 569 |
| 7.44.2.125nppiYCbCrToRGB_8u_P3C3R . . . . .        | 569 |
| 7.44.2.126nppiYCbCrToRGB_8u_P3C4R . . . . .        | 570 |
| 7.44.2.127nppiYCbCrToRGB_8u_P3R . . . . .          | 570 |
| 7.44.2.128nppiYCCToRGB_8u_AC4R . . . . .           | 571 |
| 7.44.2.129nppiYCCToRGB_8u_C3R . . . . .            | 571 |
| 7.44.2.130nppiYCrCb420ToRGB_8u_P3C4R . . . . .     | 571 |
| 7.44.2.131nppiYCrCb422ToRGB_8u_C2C3R . . . . .     | 572 |
| 7.44.2.132nppiYCrCb422ToRGB_8u_C2P3R . . . . .     | 572 |
| 7.44.2.133nppiYUV420ToBGR_8u_P3C3R . . . . .       | 572 |
| 7.44.2.134nppiYUV420ToRGB_8u_P3AC4R . . . . .      | 573 |
| 7.44.2.135nppiYUV420ToRGB_8u_P3C3R . . . . .       | 573 |
| 7.44.2.136nppiYUV420ToRGB_8u_P3R . . . . .         | 573 |
| 7.44.2.137nppiYUV422ToRGB_8u_C2C3R . . . . .       | 574 |
| 7.44.2.138nppiYUV422ToRGB_8u_P3AC4R . . . . .      | 574 |
| 7.44.2.139nppiYUV422ToRGB_8u_P3C3R . . . . .       | 574 |
| 7.44.2.140nppiYUV422ToRGB_8u_P3R . . . . .         | 575 |
| 7.44.2.141nppiYUVToBGR_8u_AC4R . . . . .           | 575 |
| 7.44.2.142nppiYUVToBGR_8u_C3R . . . . .            | 575 |
| 7.44.2.143nppiYUVToBGR_8u_P3C3R . . . . .          | 576 |
| 7.44.2.144nppiYUVToBGR_8u_P3R . . . . .            | 576 |
| 7.44.2.145nppiYUVToRGB_8u_AC4R . . . . .           | 576 |

|            |   |     |
|------------|---|-----|
| 7.44.2.146 | <a href="#">nppiYUVToRGB_8u_C3R</a>             | 577 |
| 7.44.2.147 | <a href="#">nppiYUVToRGB_8u_P3C3R</a>           | 577 |
| 7.44.2.148 | <a href="#">nppiYUVToRGB_8u_P3R</a>             | 577 |
| 7.45       | Color Sampling Format Conversion                | 578 |
| 7.45.1     | Detailed Description                            | 585 |
| 7.45.2     | Function Documentation                          | 585 |
| 7.45.2.1   | <a href="#">nppiCbYCr422ToYCbCr411_8u_C2P3R</a> | 585 |
| 7.45.2.2   | <a href="#">nppiCbYCr422ToYCbCr420_8u_C2P2R</a> | 586 |
| 7.45.2.3   | <a href="#">nppiCbYCr422ToYCbCr420_8u_C2P3R</a> | 586 |
| 7.45.2.4   | <a href="#">nppiCbYCr422ToYCbCr422_8u_C2P3R</a> | 587 |
| 7.45.2.5   | <a href="#">nppiCbYCr422ToYCbCr422_8u_C2R</a>   | 587 |
| 7.45.2.6   | <a href="#">nppiCbYCr422ToYCrCb420_8u_C2P3R</a> | 587 |
| 7.45.2.7   | <a href="#">nppiYCbCr411_8u_P2P3R</a>           | 588 |
| 7.45.2.8   | <a href="#">nppiYCbCr411_8u_P3P2R</a>           | 588 |
| 7.45.2.9   | <a href="#">nppiYCbCr411ToYCbCr420_8u_P2P3R</a> | 589 |
| 7.45.2.10  | <a href="#">nppiYCbCr411ToYCbCr420_8u_P3P2R</a> | 589 |
| 7.45.2.11  | <a href="#">nppiYCbCr411ToYCbCr420_8u_P3R</a>   | 589 |
| 7.45.2.12  | <a href="#">nppiYCbCr411ToYCbCr422_8u_P2C2R</a> | 590 |
| 7.45.2.13  | <a href="#">nppiYCbCr411ToYCbCr422_8u_P2P3R</a> | 590 |
| 7.45.2.14  | <a href="#">nppiYCbCr411ToYCbCr422_8u_P3C2R</a> | 591 |
| 7.45.2.15  | <a href="#">nppiYCbCr411ToYCbCr422_8u_P3R</a>   | 591 |
| 7.45.2.16  | <a href="#">nppiYCbCr411ToYCrCb420_8u_P2P3R</a> | 591 |
| 7.45.2.17  | <a href="#">nppiYCbCr411ToYCrCb422_8u_P3C2R</a> | 592 |
| 7.45.2.18  | <a href="#">nppiYCbCr411ToYCrCb422_8u_P3R</a>   | 592 |
| 7.45.2.19  | <a href="#">nppiYCbCr420_8u_P2P3R</a>           | 593 |
| 7.45.2.20  | <a href="#">nppiYCbCr420_8u_P3P2R</a>           | 593 |
| 7.45.2.21  | <a href="#">nppiYCbCr420ToCbYCr422_8u_P2C2R</a> | 593 |
| 7.45.2.22  | <a href="#">nppiYCbCr420ToYCbCr411_8u_P2P3R</a> | 594 |
| 7.45.2.23  | <a href="#">nppiYCbCr420ToYCbCr411_8u_P3P2R</a> | 594 |
| 7.45.2.24  | <a href="#">nppiYCbCr420ToYCbCr422_8u_P2C2R</a> | 595 |
| 7.45.2.25  | <a href="#">nppiYCbCr420ToYCbCr422_8u_P2P3R</a> | 595 |
| 7.45.2.26  | <a href="#">nppiYCbCr420ToYCbCr422_8u_P3R</a>   | 596 |
| 7.45.2.27  | <a href="#">nppiYCbCr420ToYCrCb420_8u_P2P3R</a> | 596 |
| 7.45.2.28  | <a href="#">nppiYCbCr422_8u_C2P3R</a>           | 596 |
| 7.45.2.29  | <a href="#">nppiYCbCr422_8u_P3C2R</a>           | 597 |
| 7.45.2.30  | <a href="#">nppiYCbCr422ToCbYCr422_8u_C2R</a>   | 597 |

|           |   |     |
|-----------|---|-----|
| 7.45.2.31 | <a href="#">nppiYCbCr422ToYCbCr411_8u_C2P2R</a> | 598 |
| 7.45.2.32 | <a href="#">nppiYCbCr422ToYCbCr411_8u_C2P3R</a> | 598 |
| 7.45.2.33 | <a href="#">nppiYCbCr422ToYCbCr411_8u_P3P2R</a> | 598 |
| 7.45.2.34 | <a href="#">nppiYCbCr422ToYCbCr411_8u_P3R</a>   | 599 |
| 7.45.2.35 | <a href="#">nppiYCbCr422ToYCbCr420_8u_C2P2R</a> | 599 |
| 7.45.2.36 | <a href="#">nppiYCbCr422ToYCbCr420_8u_C2P3R</a> | 600 |
| 7.45.2.37 | <a href="#">nppiYCbCr422ToYCbCr420_8u_P3P2R</a> | 600 |
| 7.45.2.38 | <a href="#">nppiYCbCr422ToYCbCr420_8u_P3R</a>   | 601 |
| 7.45.2.39 | <a href="#">nppiYCbCr422ToYCrCb420_8u_C2P3R</a> | 601 |
| 7.45.2.40 | <a href="#">nppiYCbCr422ToYCrCb422_8u_C2R</a>   | 601 |
| 7.45.2.41 | <a href="#">nppiYCbCr422ToYCrCb422_8u_P3C2R</a> | 602 |
| 7.45.2.42 | <a href="#">nppiYCrCb420ToCbYCr422_8u_P3C2R</a> | 602 |
| 7.45.2.43 | <a href="#">nppiYCrCb420ToYCbCr411_8u_P3P2R</a> | 603 |
| 7.45.2.44 | <a href="#">nppiYCrCb420ToYCbCr420_8u_P3P2R</a> | 603 |
| 7.45.2.45 | <a href="#">nppiYCrCb420ToYCbCr422_8u_P3C2R</a> | 604 |
| 7.45.2.46 | <a href="#">nppiYCrCb420ToYCbCr422_8u_P3R</a>   | 604 |
| 7.45.2.47 | <a href="#">nppiYCrCb422ToYCbCr411_8u_C2P3R</a> | 604 |
| 7.45.2.48 | <a href="#">nppiYCrCb422ToYCbCr420_8u_C2P3R</a> | 605 |
| 7.45.2.49 | <a href="#">nppiYCrCb422ToYCbCr422_8u_C2P3R</a> | 605 |
| 7.46      | <a href="#">Color Gamma Correction</a>          | 606 |
| 7.46.1    | <a href="#">Detailed Description</a>            | 607 |
| 7.46.2    | <a href="#">Function Documentation</a>          | 607 |
| 7.46.2.1  | <a href="#">nppiGammaFwd_8u_AC4IR</a>           | 607 |
| 7.46.2.2  | <a href="#">nppiGammaFwd_8u_AC4R</a>            | 607 |
| 7.46.2.3  | <a href="#">nppiGammaFwd_8u_C3IR</a>            | 608 |
| 7.46.2.4  | <a href="#">nppiGammaFwd_8u_C3R</a>             | 608 |
| 7.46.2.5  | <a href="#">nppiGammaFwd_8u_IP3R</a>            | 608 |
| 7.46.2.6  | <a href="#">nppiGammaFwd_8u_P3R</a>             | 609 |
| 7.46.2.7  | <a href="#">nppiGammaInv_8u_AC4IR</a>           | 609 |
| 7.46.2.8  | <a href="#">nppiGammaInv_8u_AC4R</a>            | 609 |
| 7.46.2.9  | <a href="#">nppiGammaInv_8u_C3IR</a>            | 610 |
| 7.46.2.10 | <a href="#">nppiGammaInv_8u_C3R</a>             | 610 |
| 7.46.2.11 | <a href="#">nppiGammaInv_8u_IP3R</a>            | 610 |
| 7.46.2.12 | <a href="#">nppiGammaInv_8u_P3R</a>             | 611 |
| 7.47      | <a href="#">Complement Color Key</a>            | 612 |
| 7.47.1    | <a href="#">Detailed Description</a>            | 612 |

|           |                               |     |
|-----------|-------------------------------|-----|
| 7.47.2    | Function Documentation        | 612 |
| 7.47.2.1  | nppiAlphaCompColorKey_8u_AC4R | 612 |
| 7.47.2.2  | nppiCompColorKey_8u_C1R       | 613 |
| 7.47.2.3  | nppiCompColorKey_8u_C3R       | 613 |
| 7.47.2.4  | nppiCompColorKey_8u_C4R       | 614 |
| 7.48      | Color Processing              | 615 |
| 7.48.1    | Detailed Description          | 627 |
| 7.48.2    | Function Documentation        | 628 |
| 7.48.2.1  | nppiColorTwist32f_16s_AC4IR   | 628 |
| 7.48.2.2  | nppiColorTwist32f_16s_AC4R    | 628 |
| 7.48.2.3  | nppiColorTwist32f_16s_C3IR    | 628 |
| 7.48.2.4  | nppiColorTwist32f_16s_C3R     | 629 |
| 7.48.2.5  | nppiColorTwist32f_16s_IP3R    | 629 |
| 7.48.2.6  | nppiColorTwist32f_16s_P3R     | 630 |
| 7.48.2.7  | nppiColorTwist32f_16u_AC4IR   | 630 |
| 7.48.2.8  | nppiColorTwist32f_16u_AC4R    | 630 |
| 7.48.2.9  | nppiColorTwist32f_16u_C3IR    | 631 |
| 7.48.2.10 | nppiColorTwist32f_16u_C3R     | 631 |
| 7.48.2.11 | nppiColorTwist32f_16u_IP3R    | 632 |
| 7.48.2.12 | nppiColorTwist32f_16u_P3R     | 632 |
| 7.48.2.13 | nppiColorTwist32f_8s_AC4IR    | 632 |
| 7.48.2.14 | nppiColorTwist32f_8s_AC4R     | 633 |
| 7.48.2.15 | nppiColorTwist32f_8s_C3IR     | 633 |
| 7.48.2.16 | nppiColorTwist32f_8s_C3R      | 633 |
| 7.48.2.17 | nppiColorTwist32f_8s_C4IR     | 634 |
| 7.48.2.18 | nppiColorTwist32f_8s_C4R      | 634 |
| 7.48.2.19 | nppiColorTwist32f_8s_IP3R     | 635 |
| 7.48.2.20 | nppiColorTwist32f_8s_P3R      | 635 |
| 7.48.2.21 | nppiColorTwist32f_8u_AC4IR    | 635 |
| 7.48.2.22 | nppiColorTwist32f_8u_AC4R     | 636 |
| 7.48.2.23 | nppiColorTwist32f_8u_C3IR     | 636 |
| 7.48.2.24 | nppiColorTwist32f_8u_C3R      | 636 |
| 7.48.2.25 | nppiColorTwist32f_8u_C4IR     | 637 |
| 7.48.2.26 | nppiColorTwist32f_8u_C4R      | 637 |
| 7.48.2.27 | nppiColorTwist32f_8u_IP3R     | 638 |
| 7.48.2.28 | nppiColorTwist32f_8u_P3R      | 638 |

|  |     |
|--|-----|
| 7.48.2.29 nppiColorTwist_32f_AC4IR . . . . . | 638 |
| 7.48.2.30 nppiColorTwist_32f_AC4R . . . . .  | 639 |
| 7.48.2.31 nppiColorTwist_32f_C3IR . . . . .  | 639 |
| 7.48.2.32 nppiColorTwist_32f_C3R . . . . .   | 639 |
| 7.48.2.33 nppiColorTwist_32f_C4IR . . . . .  | 640 |
| 7.48.2.34 nppiColorTwist_32f_C4R . . . . .   | 640 |
| 7.48.2.35 nppiColorTwist_32f_IP3R . . . . .  | 641 |
| 7.48.2.36 nppiColorTwist_32f_P3R . . . . .   | 641 |
| 7.48.2.37 nppiLUT_16s_AC4IR . . . . .        | 641 |
| 7.48.2.38 nppiLUT_16s_AC4R . . . . .         | 642 |
| 7.48.2.39 nppiLUT_16s_C1IR . . . . .         | 642 |
| 7.48.2.40 nppiLUT_16s_C1R . . . . .          | 643 |
| 7.48.2.41 nppiLUT_16s_C3IR . . . . .         | 643 |
| 7.48.2.42 nppiLUT_16s_C3R . . . . .          | 644 |
| 7.48.2.43 nppiLUT_16s_C4IR . . . . .         | 644 |
| 7.48.2.44 nppiLUT_16s_C4R . . . . .          | 645 |
| 7.48.2.45 nppiLUT_16u_AC4IR . . . . .        | 646 |
| 7.48.2.46 nppiLUT_16u_AC4R . . . . .         | 646 |
| 7.48.2.47 nppiLUT_16u_C1IR . . . . .         | 647 |
| 7.48.2.48 nppiLUT_16u_C1R . . . . .          | 647 |
| 7.48.2.49 nppiLUT_16u_C3IR . . . . .         | 648 |
| 7.48.2.50 nppiLUT_16u_C3R . . . . .          | 648 |
| 7.48.2.51 nppiLUT_16u_C4IR . . . . .         | 649 |
| 7.48.2.52 nppiLUT_16u_C4R . . . . .          | 649 |
| 7.48.2.53 nppiLUT_32f_AC4IR . . . . .        | 650 |
| 7.48.2.54 nppiLUT_32f_AC4R . . . . .         | 650 |
| 7.48.2.55 nppiLUT_32f_C1IR . . . . .         | 651 |
| 7.48.2.56 nppiLUT_32f_C1R . . . . .          | 651 |
| 7.48.2.57 nppiLUT_32f_C3IR . . . . .         | 652 |
| 7.48.2.58 nppiLUT_32f_C3R . . . . .          | 652 |
| 7.48.2.59 nppiLUT_32f_C4IR . . . . .         | 653 |
| 7.48.2.60 nppiLUT_32f_C4R . . . . .          | 653 |
| 7.48.2.61 nppiLUT_8u_AC4IR . . . . .         | 654 |
| 7.48.2.62 nppiLUT_8u_AC4R . . . . .          | 654 |
| 7.48.2.63 nppiLUT_8u_C1IR . . . . .          | 655 |
| 7.48.2.64 nppiLUT_8u_C1R . . . . .           | 655 |

|   |     |
|---|-----|
| 7.48.2.65 nppiLUT_8u_C3IR . . . . .         | 656 |
| 7.48.2.66 nppiLUT_8u_C3R . . . . .          | 656 |
| 7.48.2.67 nppiLUT_8u_C4IR . . . . .         | 657 |
| 7.48.2.68 nppiLUT_8u_C4R . . . . .          | 657 |
| 7.48.2.69 nppiLUT_Cubic_16s_AC4IR . . . . . | 658 |
| 7.48.2.70 nppiLUT_Cubic_16s_AC4R . . . . .  | 658 |
| 7.48.2.71 nppiLUT_Cubic_16s_C1IR . . . . .  | 659 |
| 7.48.2.72 nppiLUT_Cubic_16s_C1R . . . . .   | 659 |
| 7.48.2.73 nppiLUT_Cubic_16s_C3IR . . . . .  | 660 |
| 7.48.2.74 nppiLUT_Cubic_16s_C3R . . . . .   | 660 |
| 7.48.2.75 nppiLUT_Cubic_16s_C4IR . . . . .  | 661 |
| 7.48.2.76 nppiLUT_Cubic_16s_C4R . . . . .   | 661 |
| 7.48.2.77 nppiLUT_Cubic_16u_AC4IR . . . . . | 662 |
| 7.48.2.78 nppiLUT_Cubic_16u_AC4R . . . . .  | 662 |
| 7.48.2.79 nppiLUT_Cubic_16u_C1IR . . . . .  | 663 |
| 7.48.2.80 nppiLUT_Cubic_16u_C1R . . . . .   | 663 |
| 7.48.2.81 nppiLUT_Cubic_16u_C3IR . . . . .  | 664 |
| 7.48.2.82 nppiLUT_Cubic_16u_C3R . . . . .   | 664 |
| 7.48.2.83 nppiLUT_Cubic_16u_C4IR . . . . .  | 665 |
| 7.48.2.84 nppiLUT_Cubic_16u_C4R . . . . .   | 665 |
| 7.48.2.85 nppiLUT_Cubic_32f_AC4IR . . . . . | 666 |
| 7.48.2.86 nppiLUT_Cubic_32f_AC4R . . . . .  | 666 |
| 7.48.2.87 nppiLUT_Cubic_32f_C1IR . . . . .  | 667 |
| 7.48.2.88 nppiLUT_Cubic_32f_C1R . . . . .   | 667 |
| 7.48.2.89 nppiLUT_Cubic_32f_C3IR . . . . .  | 668 |
| 7.48.2.90 nppiLUT_Cubic_32f_C3R . . . . .   | 668 |
| 7.48.2.91 nppiLUT_Cubic_32f_C4IR . . . . .  | 669 |
| 7.48.2.92 nppiLUT_Cubic_32f_C4R . . . . .   | 669 |
| 7.48.2.93 nppiLUT_Cubic_8u_AC4IR . . . . .  | 670 |
| 7.48.2.94 nppiLUT_Cubic_8u_AC4R . . . . .   | 670 |
| 7.48.2.95 nppiLUT_Cubic_8u_C1IR . . . . .   | 671 |
| 7.48.2.96 nppiLUT_Cubic_8u_C1R . . . . .    | 671 |
| 7.48.2.97 nppiLUT_Cubic_8u_C3IR . . . . .   | 672 |
| 7.48.2.98 nppiLUT_Cubic_8u_C3R . . . . .    | 672 |
| 7.48.2.99 nppiLUT_Cubic_8u_C4IR . . . . .   | 673 |
| 7.48.2.100 nppiLUT_Cubic_8u_C4R . . . . .   | 673 |

|  |     |
|--|-----|
| 7.48.2.101nppiLUT_Linear_16s_AC4IR . . . . .   | 674 |
| 7.48.2.102nppiLUT_Linear_16s_AC4R . . . . .    | 674 |
| 7.48.2.103nppiLUT_Linear_16s_C1IR . . . . .    | 675 |
| 7.48.2.104nppiLUT_Linear_16s_C1R . . . . .     | 675 |
| 7.48.2.105nppiLUT_Linear_16s_C3IR . . . . .    | 676 |
| 7.48.2.106nppiLUT_Linear_16s_C3R . . . . .     | 676 |
| 7.48.2.107nppiLUT_Linear_16s_C4IR . . . . .    | 677 |
| 7.48.2.108nppiLUT_Linear_16s_C4R . . . . .     | 677 |
| 7.48.2.109nppiLUT_Linear_16u_AC4IR . . . . .   | 678 |
| 7.48.2.110nppiLUT_Linear_16u_AC4R . . . . .    | 678 |
| 7.48.2.111nppiLUT_Linear_16u_C1IR . . . . .    | 679 |
| 7.48.2.112nppiLUT_Linear_16u_C1R . . . . .     | 679 |
| 7.48.2.113nppiLUT_Linear_16u_C3IR . . . . .    | 680 |
| 7.48.2.114nppiLUT_Linear_16u_C3R . . . . .     | 680 |
| 7.48.2.115nppiLUT_Linear_16u_C4IR . . . . .    | 681 |
| 7.48.2.116nppiLUT_Linear_16u_C4R . . . . .     | 681 |
| 7.48.2.117nppiLUT_Linear_32f_AC4IR . . . . .   | 682 |
| 7.48.2.118nppiLUT_Linear_32f_AC4R . . . . .    | 682 |
| 7.48.2.119nppiLUT_Linear_32f_C1IR . . . . .    | 683 |
| 7.48.2.120nppiLUT_Linear_32f_C1R . . . . .     | 683 |
| 7.48.2.121nppiLUT_Linear_32f_C3IR . . . . .    | 684 |
| 7.48.2.122nppiLUT_Linear_32f_C3R . . . . .     | 684 |
| 7.48.2.123nppiLUT_Linear_32f_C4IR . . . . .    | 685 |
| 7.48.2.124nppiLUT_Linear_32f_C4R . . . . .     | 685 |
| 7.48.2.125nppiLUT_Linear_8u_AC4IR . . . . .    | 686 |
| 7.48.2.126nppiLUT_Linear_8u_AC4R . . . . .     | 686 |
| 7.48.2.127nppiLUT_Linear_8u_C1IR . . . . .     | 687 |
| 7.48.2.128nppiLUT_Linear_8u_C1R . . . . .      | 687 |
| 7.48.2.129nppiLUT_Linear_8u_C3IR . . . . .     | 688 |
| 7.48.2.130nppiLUT_Linear_8u_C3R . . . . .      | 688 |
| 7.48.2.131nppiLUT_Linear_8u_C4IR . . . . .     | 689 |
| 7.48.2.132nppiLUT_Linear_8u_C4R . . . . .      | 689 |
| 7.48.2.133nppiLUT_Trilinear_8u_AC4IR . . . . . | 690 |
| 7.48.2.134nppiLUT_Trilinear_8u_AC4R . . . . .  | 691 |
| 7.48.2.135nppiLUT_Trilinear_8u_C4R . . . . .   | 691 |
| 7.48.2.136nppiLUTPalette_16u24u_C1R . . . . .  | 692 |



|            |  |     |
|------------|--|-----|
| 7.48.2.137 | <a href="#">nppiLUTPalette_16u32u_C1R</a>                | 693 |
| 7.48.2.138 | <a href="#">nppiLUTPalette_16u8u_C1R</a>                 | 693 |
| 7.48.2.139 | <a href="#">nppiLUTPalette_16u_AC4R</a>                  | 694 |
| 7.48.2.140 | <a href="#">nppiLUTPalette_16u_C1R</a>                   | 694 |
| 7.48.2.141 | <a href="#">nppiLUTPalette_16u_C3R</a>                   | 695 |
| 7.48.2.142 | <a href="#">nppiLUTPalette_16u_C4R</a>                   | 695 |
| 7.48.2.143 | <a href="#">nppiLUTPalette_8u24u_C1R</a>                 | 696 |
| 7.48.2.144 | <a href="#">nppiLUTPalette_8u32u_C1R</a>                 | 696 |
| 7.48.2.145 | <a href="#">nppiLUTPalette_8u_AC4R</a>                   | 697 |
| 7.48.2.146 | <a href="#">nppiLUTPalette_8u_C1R</a>                    | 697 |
| 7.48.2.147 | <a href="#">nppiLUTPalette_8u_C3R</a>                    | 698 |
| 7.48.2.148 | <a href="#">nppiLUTPalette_8u_C4R</a>                    | 698 |
| 7.48.2.149 | <a href="#">nppiLUTPaletteSwap_16u_C3A0C4R</a>           | 699 |
| 7.48.2.150 | <a href="#">nppiLUTPaletteSwap_8u_C3A0C4R</a>            | 699 |
| 7.49       | Compression  | 701 |
| 7.49.1     | Detailed Description                                     | 701 |
| 7.49.2     | Typedef Documentation                                    | 702 |
| 7.49.2.1   | <a href="#">NppiDecodeHuffmanSpec</a>                    | 702 |
| 7.49.3     | Function Documentation                                   | 702 |
| 7.49.3.1   | <a href="#">nppiDecodeHuffmanScanHost_JPEG_8u16s_P1R</a> | 702 |
| 7.49.3.2   | <a href="#">nppiDecodeHuffmanScanHost_JPEG_8u16s_P3R</a> | 702 |
| 7.49.3.3   | <a href="#">nppiDecodeHuffmanSpecFreeHost_JPEG</a>       | 703 |
| 7.49.3.4   | <a href="#">nppiDecodeHuffmanSpecGetBufSize_JPEG</a>     | 703 |
| 7.49.3.5   | <a href="#">nppiDecodeHuffmanSpecInitAllocHost_JPEG</a>  | 704 |
| 7.49.3.6   | <a href="#">nppiDecodeHuffmanSpecInitHost_JPEG</a>       | 704 |
| 7.50       | Quantization Functions                                   | 705 |
| 7.50.1     | Typedef Documentation                                    | 706 |
| 7.50.1.1   | <a href="#">NppiDCTState</a>                             | 706 |
| 7.50.2     | Function Documentation                                   | 706 |
| 7.50.2.1   | <a href="#">nppiDCTFree</a>                              | 706 |
| 7.50.2.2   | <a href="#">nppiDCTInitAlloc</a>                         | 706 |
| 7.50.2.3   | <a href="#">nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R</a>      | 706 |
| 7.50.2.4   | <a href="#">nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_NEW</a>  | 707 |
| 7.50.2.5   | <a href="#">nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R</a>      | 707 |
| 7.50.2.6   | <a href="#">nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_NEW</a>  | 708 |
| 7.50.2.7   | <a href="#">nppiQuantFwdRawTableInit_JPEG_8u</a>         | 709 |

|           |   |     |
|-----------|---|-----|
| 7.50.2.8  | <code>nppiQuantFwdTableInit_JPEG_8u16u</code> | 709 |
| 7.50.2.9  | <code>nppiQuantInvTableInit_JPEG_8u16u</code> | 709 |
| 7.51      | Labeling and Segmentation                     | 711 |
| 7.51.1    | Detailed Description                          | 711 |
| 7.51.2    | Typedef Documentation                         | 711 |
| 7.51.2.1  | <code>NppiGraphcutState</code>                | 711 |
| 7.52      | GraphCut                                      | 712 |
| 7.52.1    | Function Documentation                        | 713 |
| 7.52.1.1  | <code>nppiGraphcut8_32f8u</code>              | 713 |
| 7.52.1.2  | <code>nppiGraphcut8_32s8u</code>              | 714 |
| 7.52.1.3  | <code>nppiGraphcut8GetSize</code>             | 714 |
| 7.52.1.4  | <code>nppiGraphcut8InitAlloc</code>           | 715 |
| 7.52.1.5  | <code>nppiGraphcut_32f8u</code>               | 715 |
| 7.52.1.6  | <code>nppiGraphcut_32s8u</code>               | 716 |
| 7.52.1.7  | <code>nppiGraphcutFree</code>                 | 717 |
| 7.52.1.8  | <code>nppiGraphcutGetSize</code>              | 717 |
| 7.52.1.9  | <code>nppiGraphcutInitAlloc</code>            | 718 |
| 7.53      | Data Exchange and Initialization              | 719 |
| 7.53.1    | Detailed Description                          | 719 |
| 7.54      | Set   | 720 |
| 7.54.1    | Detailed Description                          | 726 |
| 7.54.2    | Function Documentation                        | 726 |
| 7.54.2.1  | <code>nppiSet_16s_AC4MR</code>                | 726 |
| 7.54.2.2  | <code>nppiSet_16s_AC4R</code>                 | 727 |
| 7.54.2.3  | <code>nppiSet_16s_C1MR</code>                 | 727 |
| 7.54.2.4  | <code>nppiSet_16s_C1R</code>                  | 727 |
| 7.54.2.5  | <code>nppiSet_16s_C2R</code>                  | 728 |
| 7.54.2.6  | <code>nppiSet_16s_C3CR</code>                 | 728 |
| 7.54.2.7  | <code>nppiSet_16s_C3MR</code>                 | 728 |
| 7.54.2.8  | <code>nppiSet_16s_C3R</code>                  | 729 |
| 7.54.2.9  | <code>nppiSet_16s_C4CR</code>                 | 729 |
| 7.54.2.10 | <code>nppiSet_16s_C4MR</code>                 | 729 |
| 7.54.2.11 | <code>nppiSet_16s_C4R</code>                  | 730 |
| 7.54.2.12 | <code>nppiSet_16sc_AC4R</code>                | 730 |
| 7.54.2.13 | <code>nppiSet_16sc_C1R</code>                 | 730 |
| 7.54.2.14 | <code>nppiSet_16sc_C2R</code>                 | 731 |

|                                       |     |
|---------------------------------------|-----|
| 7.54.2.15 nppiSet_16sc_C3R . . . . .  | 731 |
| 7.54.2.16 nppiSet_16sc_C4R . . . . .  | 731 |
| 7.54.2.17 nppiSet_16u_AC4MR . . . . . | 732 |
| 7.54.2.18 nppiSet_16u_AC4R . . . . .  | 732 |
| 7.54.2.19 nppiSet_16u_C1MR . . . . .  | 732 |
| 7.54.2.20 nppiSet_16u_C1R . . . . .   | 733 |
| 7.54.2.21 nppiSet_16u_C2R . . . . .   | 733 |
| 7.54.2.22 nppiSet_16u_C3CR . . . . .  | 733 |
| 7.54.2.23 nppiSet_16u_C3MR . . . . .  | 734 |
| 7.54.2.24 nppiSet_16u_C3R . . . . .   | 734 |
| 7.54.2.25 nppiSet_16u_C4CR . . . . .  | 734 |
| 7.54.2.26 nppiSet_16u_C4MR . . . . .  | 735 |
| 7.54.2.27 nppiSet_16u_C4R . . . . .   | 735 |
| 7.54.2.28 nppiSet_32f_AC4MR . . . . . | 735 |
| 7.54.2.29 nppiSet_32f_AC4R . . . . .  | 736 |
| 7.54.2.30 nppiSet_32f_C1MR . . . . .  | 736 |
| 7.54.2.31 nppiSet_32f_C1R . . . . .   | 737 |
| 7.54.2.32 nppiSet_32f_C2R . . . . .   | 737 |
| 7.54.2.33 nppiSet_32f_C3CR . . . . .  | 737 |
| 7.54.2.34 nppiSet_32f_C3MR . . . . .  | 738 |
| 7.54.2.35 nppiSet_32f_C3R . . . . .   | 738 |
| 7.54.2.36 nppiSet_32f_C4CR . . . . .  | 738 |
| 7.54.2.37 nppiSet_32f_C4MR . . . . .  | 739 |
| 7.54.2.38 nppiSet_32f_C4R . . . . .   | 739 |
| 7.54.2.39 nppiSet_32fc_AC4R . . . . . | 739 |
| 7.54.2.40 nppiSet_32fc_C1R . . . . .  | 740 |
| 7.54.2.41 nppiSet_32fc_C2R . . . . .  | 740 |
| 7.54.2.42 nppiSet_32fc_C3R . . . . .  | 740 |
| 7.54.2.43 nppiSet_32fc_C4R . . . . .  | 741 |
| 7.54.2.44 nppiSet_32s_AC4MR . . . . . | 741 |
| 7.54.2.45 nppiSet_32s_AC4R . . . . .  | 741 |
| 7.54.2.46 nppiSet_32s_C1MR . . . . .  | 742 |
| 7.54.2.47 nppiSet_32s_C1R . . . . .   | 742 |
| 7.54.2.48 nppiSet_32s_C2R . . . . .   | 742 |
| 7.54.2.49 nppiSet_32s_C3CR . . . . .  | 743 |
| 7.54.2.50 nppiSet_32s_C3MR . . . . .  | 743 |

|           |                                 |     |
|-----------|---------------------------------|-----|
| 7.54.2.51 | <code>nppiSet_32s_C3R</code>    | 743 |
| 7.54.2.52 | <code>nppiSet_32s_C4CR</code>   | 744 |
| 7.54.2.53 | <code>nppiSet_32s_C4MR</code>   | 744 |
| 7.54.2.54 | <code>nppiSet_32s_C4R</code>    | 744 |
| 7.54.2.55 | <code>nppiSet_32sc_AC4R</code>  | 745 |
| 7.54.2.56 | <code>nppiSet_32sc_C1R</code>   | 745 |
| 7.54.2.57 | <code>nppiSet_32sc_C2R</code>   | 745 |
| 7.54.2.58 | <code>nppiSet_32sc_C3R</code>   | 746 |
| 7.54.2.59 | <code>nppiSet_32sc_C4R</code>   | 746 |
| 7.54.2.60 | <code>nppiSet_32u_AC4R</code>   | 746 |
| 7.54.2.61 | <code>nppiSet_32u_C1R</code>    | 747 |
| 7.54.2.62 | <code>nppiSet_32u_C2R</code>    | 747 |
| 7.54.2.63 | <code>nppiSet_32u_C3R</code>    | 747 |
| 7.54.2.64 | <code>nppiSet_32u_C4R</code>    | 748 |
| 7.54.2.65 | <code>nppiSet_8s_AC4R</code>    | 748 |
| 7.54.2.66 | <code>nppiSet_8s_C1R</code>     | 748 |
| 7.54.2.67 | <code>nppiSet_8s_C2R</code>     | 749 |
| 7.54.2.68 | <code>nppiSet_8s_C3R</code>     | 749 |
| 7.54.2.69 | <code>nppiSet_8s_C4R</code>     | 749 |
| 7.54.2.70 | <code>nppiSet_8u_AC4MR</code>   | 750 |
| 7.54.2.71 | <code>nppiSet_8u_AC4R</code>    | 750 |
| 7.54.2.72 | <code>nppiSet_8u_C1MR</code>    | 750 |
| 7.54.2.73 | <code>nppiSet_8u_C1R</code>     | 751 |
| 7.54.2.74 | <code>nppiSet_8u_C2R</code>     | 751 |
| 7.54.2.75 | <code>nppiSet_8u_C3CR</code>    | 751 |
| 7.54.2.76 | <code>nppiSet_8u_C3MR</code>    | 752 |
| 7.54.2.77 | <code>nppiSet_8u_C3R</code>     | 752 |
| 7.54.2.78 | <code>nppiSet_8u_C4CR</code>    | 752 |
| 7.54.2.79 | <code>nppiSet_8u_C4MR</code>    | 753 |
| 7.54.2.80 | <code>nppiSet_8u_C4R</code>     | 753 |
| 7.55      | Copy                            | 754 |
| 7.55.1    | Function Documentation          | 763 |
| 7.55.1.1  | <code>nppiCopy_16s_AC4MR</code> | 763 |
| 7.55.1.2  | <code>nppiCopy_16s_AC4R</code>  | 764 |
| 7.55.1.3  | <code>nppiCopy_16s_C1C3R</code> | 764 |
| 7.55.1.4  | <code>nppiCopy_16s_C1C4R</code> | 765 |

|           |                    |     |
|-----------|--------------------|-----|
| 7.55.1.5  | nppiCopy_16s_C1MR  | 765 |
| 7.55.1.6  | nppiCopy_16s_C1R   | 765 |
| 7.55.1.7  | nppiCopy_16s_C3C1R | 766 |
| 7.55.1.8  | nppiCopy_16s_C3CR  | 766 |
| 7.55.1.9  | nppiCopy_16s_C3MR  | 766 |
| 7.55.1.10 | nppiCopy_16s_C3P3R | 767 |
| 7.55.1.11 | nppiCopy_16s_C3R   | 767 |
| 7.55.1.12 | nppiCopy_16s_C4C1R | 767 |
| 7.55.1.13 | nppiCopy_16s_C4CR  | 768 |
| 7.55.1.14 | nppiCopy_16s_C4MR  | 768 |
| 7.55.1.15 | nppiCopy_16s_C4P4R | 768 |
| 7.55.1.16 | nppiCopy_16s_C4R   | 769 |
| 7.55.1.17 | nppiCopy_16s_P3C3R | 769 |
| 7.55.1.18 | nppiCopy_16s_P4C4R | 769 |
| 7.55.1.19 | nppiCopy_16sc_AC4R | 770 |
| 7.55.1.20 | nppiCopy_16sc_C1R  | 770 |
| 7.55.1.21 | nppiCopy_16sc_C2R  | 770 |
| 7.55.1.22 | nppiCopy_16sc_C3R  | 771 |
| 7.55.1.23 | nppiCopy_16sc_C4R  | 771 |
| 7.55.1.24 | nppiCopy_16u_AC4MR | 771 |
| 7.55.1.25 | nppiCopy_16u_AC4R  | 772 |
| 7.55.1.26 | nppiCopy_16u_C1C3R | 772 |
| 7.55.1.27 | nppiCopy_16u_C1C4R | 772 |
| 7.55.1.28 | nppiCopy_16u_C1MR  | 773 |
| 7.55.1.29 | nppiCopy_16u_C1R   | 773 |
| 7.55.1.30 | nppiCopy_16u_C3C1R | 773 |
| 7.55.1.31 | nppiCopy_16u_C3CR  | 774 |
| 7.55.1.32 | nppiCopy_16u_C3MR  | 774 |
| 7.55.1.33 | nppiCopy_16u_C3P3R | 774 |
| 7.55.1.34 | nppiCopy_16u_C3R   | 775 |
| 7.55.1.35 | nppiCopy_16u_C4C1R | 775 |
| 7.55.1.36 | nppiCopy_16u_C4CR  | 775 |
| 7.55.1.37 | nppiCopy_16u_C4MR  | 776 |
| 7.55.1.38 | nppiCopy_16u_C4P4R | 776 |
| 7.55.1.39 | nppiCopy_16u_C4R   | 776 |
| 7.55.1.40 | nppiCopy_16u_P3C3R | 777 |

|  |     |
|--|-----|
| 7.55.1.41 nppiCopy_16u_P4C4R . . . . . | 777 |
| 7.55.1.42 nppiCopy_32f_AC4MR . . . . . | 777 |
| 7.55.1.43 nppiCopy_32f_AC4R . . . . .  | 778 |
| 7.55.1.44 nppiCopy_32f_C1C3R . . . . . | 778 |
| 7.55.1.45 nppiCopy_32f_C1C4R . . . . . | 778 |
| 7.55.1.46 nppiCopy_32f_C1MR . . . . .  | 779 |
| 7.55.1.47 nppiCopy_32f_C1R . . . . .   | 779 |
| 7.55.1.48 nppiCopy_32f_C3C1R . . . . . | 779 |
| 7.55.1.49 nppiCopy_32f_C3CR . . . . .  | 780 |
| 7.55.1.50 nppiCopy_32f_C3MR . . . . .  | 780 |
| 7.55.1.51 nppiCopy_32f_C3P3R . . . . . | 780 |
| 7.55.1.52 nppiCopy_32f_C3R . . . . .   | 781 |
| 7.55.1.53 nppiCopy_32f_C4C1R . . . . . | 781 |
| 7.55.1.54 nppiCopy_32f_C4CR . . . . .  | 781 |
| 7.55.1.55 nppiCopy_32f_C4MR . . . . .  | 782 |
| 7.55.1.56 nppiCopy_32f_C4P4R . . . . . | 782 |
| 7.55.1.57 nppiCopy_32f_C4R . . . . .   | 782 |
| 7.55.1.58 nppiCopy_32f_P3C3R . . . . . | 783 |
| 7.55.1.59 nppiCopy_32f_P4C4R . . . . . | 783 |
| 7.55.1.60 nppiCopy_32fc_AC4R . . . . . | 783 |
| 7.55.1.61 nppiCopy_32fc_C1R . . . . .  | 784 |
| 7.55.1.62 nppiCopy_32fc_C2R . . . . .  | 784 |
| 7.55.1.63 nppiCopy_32fc_C3R . . . . .  | 784 |
| 7.55.1.64 nppiCopy_32fc_C4R . . . . .  | 785 |
| 7.55.1.65 nppiCopy_32s_AC4MR . . . . . | 785 |
| 7.55.1.66 nppiCopy_32s_AC4R . . . . .  | 785 |
| 7.55.1.67 nppiCopy_32s_C1C3R . . . . . | 786 |
| 7.55.1.68 nppiCopy_32s_C1C4R . . . . . | 786 |
| 7.55.1.69 nppiCopy_32s_C1MR . . . . .  | 786 |
| 7.55.1.70 nppiCopy_32s_C1R . . . . .   | 787 |
| 7.55.1.71 nppiCopy_32s_C3C1R . . . . . | 787 |
| 7.55.1.72 nppiCopy_32s_C3CR . . . . .  | 787 |
| 7.55.1.73 nppiCopy_32s_C3MR . . . . .  | 788 |
| 7.55.1.74 nppiCopy_32s_C3P3R . . . . . | 788 |
| 7.55.1.75 nppiCopy_32s_C3R . . . . .   | 788 |
| 7.55.1.76 nppiCopy_32s_C4C1R . . . . . | 789 |

|                               |     |
|-------------------------------|-----|
| 7.55.1.77 nppiCopy_32s_C4CR   | 789 |
| 7.55.1.78 nppiCopy_32s_C4MR   | 789 |
| 7.55.1.79 nppiCopy_32s_C4P4R  | 790 |
| 7.55.1.80 nppiCopy_32s_C4R    | 790 |
| 7.55.1.81 nppiCopy_32s_P3C3R  | 790 |
| 7.55.1.82 nppiCopy_32s_P4C4R  | 791 |
| 7.55.1.83 nppiCopy_32sc_AC4R  | 791 |
| 7.55.1.84 nppiCopy_32sc_C1R   | 791 |
| 7.55.1.85 nppiCopy_32sc_C2R   | 792 |
| 7.55.1.86 nppiCopy_32sc_C3R   | 792 |
| 7.55.1.87 nppiCopy_32sc_C4R   | 792 |
| 7.55.1.88 nppiCopy_8s_AC4R    | 793 |
| 7.55.1.89 nppiCopy_8s_C1R     | 793 |
| 7.55.1.90 nppiCopy_8s_C2R     | 793 |
| 7.55.1.91 nppiCopy_8s_C3R     | 794 |
| 7.55.1.92 nppiCopy_8s_C4R     | 794 |
| 7.55.1.93 nppiCopy_8u_AC4MR   | 794 |
| 7.55.1.94 nppiCopy_8u_AC4R    | 795 |
| 7.55.1.95 nppiCopy_8u_C1C3R   | 795 |
| 7.55.1.96 nppiCopy_8u_C1C4R   | 795 |
| 7.55.1.97 nppiCopy_8u_C1MR    | 796 |
| 7.55.1.98 nppiCopy_8u_C1R     | 796 |
| 7.55.1.99 nppiCopy_8u_C3C1R   | 796 |
| 7.55.1.100 nppiCopy_8u_C3CR   | 797 |
| 7.55.1.101 nppiCopy_8u_C3MR   | 797 |
| 7.55.1.102 nppiCopy_8u_C3P3R  | 797 |
| 7.55.1.103 nppiCopy_8u_C3R    | 798 |
| 7.55.1.104 nppiCopy_8u_C4C1R  | 798 |
| 7.55.1.105 nppiCopy_8u_C4CR   | 798 |
| 7.55.1.106 nppiCopy_8u_C4MR   | 799 |
| 7.55.1.107 nppiCopy_8u_C4P4R  | 799 |
| 7.55.1.108 nppiCopy_8u_C4R    | 799 |
| 7.55.1.109 nppiCopy_8u_P3C3R  | 800 |
| 7.55.1.110 nppiCopy_8u_P4C4R  | 800 |
| 7.56 Convert                  | 801 |
| 7.56.1 Function Documentation | 809 |

|           |   |     |
|-----------|---|-----|
| 7.56.1.1  | <a href="#">nppiConvert_16s16u_C1Rs</a>   | 809 |
| 7.56.1.2  | <a href="#">nppiConvert_16s32f_AC4R</a>   | 809 |
| 7.56.1.3  | <a href="#">nppiConvert_16s32f_C1R</a>    | 810 |
| 7.56.1.4  | <a href="#">nppiConvert_16s32f_C3R</a>    | 810 |
| 7.56.1.5  | <a href="#">nppiConvert_16s32f_C4R</a>    | 810 |
| 7.56.1.6  | <a href="#">nppiConvert_16s32s_AC4R</a>   | 811 |
| 7.56.1.7  | <a href="#">nppiConvert_16s32s_C1R</a>    | 811 |
| 7.56.1.8  | <a href="#">nppiConvert_16s32s_C3R</a>    | 811 |
| 7.56.1.9  | <a href="#">nppiConvert_16s32s_C4R</a>    | 812 |
| 7.56.1.10 | <a href="#">nppiConvert_16s32u_C1Rs</a>   | 812 |
| 7.56.1.11 | <a href="#">nppiConvert_16s8s_C1RSfs</a>  | 812 |
| 7.56.1.12 | <a href="#">nppiConvert_16s8u_AC4R</a>    | 813 |
| 7.56.1.13 | <a href="#">nppiConvert_16s8u_C1R</a>     | 813 |
| 7.56.1.14 | <a href="#">nppiConvert_16s8u_C3R</a>     | 814 |
| 7.56.1.15 | <a href="#">nppiConvert_16s8u_C4R</a>     | 814 |
| 7.56.1.16 | <a href="#">nppiConvert_16u16s_C1RSfs</a> | 814 |
| 7.56.1.17 | <a href="#">nppiConvert_16u32f_AC4R</a>   | 815 |
| 7.56.1.18 | <a href="#">nppiConvert_16u32f_C1R</a>    | 815 |
| 7.56.1.19 | <a href="#">nppiConvert_16u32f_C3R</a>    | 815 |
| 7.56.1.20 | <a href="#">nppiConvert_16u32f_C4R</a>    | 816 |
| 7.56.1.21 | <a href="#">nppiConvert_16u32s_AC4R</a>   | 816 |
| 7.56.1.22 | <a href="#">nppiConvert_16u32s_C1R</a>    | 816 |
| 7.56.1.23 | <a href="#">nppiConvert_16u32s_C3R</a>    | 817 |
| 7.56.1.24 | <a href="#">nppiConvert_16u32s_C4R</a>    | 817 |
| 7.56.1.25 | <a href="#">nppiConvert_16u32u_C1R</a>    | 817 |
| 7.56.1.26 | <a href="#">nppiConvert_16u8s_C1RSfs</a>  | 818 |
| 7.56.1.27 | <a href="#">nppiConvert_16u8u_AC4R</a>    | 818 |
| 7.56.1.28 | <a href="#">nppiConvert_16u8u_C1R</a>     | 818 |
| 7.56.1.29 | <a href="#">nppiConvert_16u8u_C3R</a>     | 819 |
| 7.56.1.30 | <a href="#">nppiConvert_16u8u_C4R</a>     | 819 |
| 7.56.1.31 | <a href="#">nppiConvert_32f16s_AC4R</a>   | 819 |
| 7.56.1.32 | <a href="#">nppiConvert_32f16s_C1R</a>    | 820 |
| 7.56.1.33 | <a href="#">nppiConvert_32f16s_C1RSfs</a> | 820 |
| 7.56.1.34 | <a href="#">nppiConvert_32f16s_C3R</a>    | 820 |
| 7.56.1.35 | <a href="#">nppiConvert_32f16s_C4R</a>    | 821 |
| 7.56.1.36 | <a href="#">nppiConvert_32f16u_AC4R</a>   | 821 |



|   |     |
|---|-----|
| 7.56.1.37 nppiConvert_32f16u_C1R . . . . .    | 822 |
| 7.56.1.38 nppiConvert_32f16u_C1RSfs . . . . . | 822 |
| 7.56.1.39 nppiConvert_32f16u_C3R . . . . .    | 822 |
| 7.56.1.40 nppiConvert_32f16u_C4R . . . . .    | 823 |
| 7.56.1.41 nppiConvert_32f32s_C1RSfs . . . . . | 823 |
| 7.56.1.42 nppiConvert_32f32u_C1RSfs . . . . . | 824 |
| 7.56.1.43 nppiConvert_32f8s_AC4R . . . . .    | 824 |
| 7.56.1.44 nppiConvert_32f8s_C1R . . . . .     | 824 |
| 7.56.1.45 nppiConvert_32f8s_C1RSfs . . . . .  | 825 |
| 7.56.1.46 nppiConvert_32f8s_C3R . . . . .     | 825 |
| 7.56.1.47 nppiConvert_32f8s_C4R . . . . .     | 826 |
| 7.56.1.48 nppiConvert_32f8u_AC4R . . . . .    | 826 |
| 7.56.1.49 nppiConvert_32f8u_C1R . . . . .     | 826 |
| 7.56.1.50 nppiConvert_32f8u_C1RSfs . . . . .  | 827 |
| 7.56.1.51 nppiConvert_32f8u_C3R . . . . .     | 827 |
| 7.56.1.52 nppiConvert_32f8u_C4R . . . . .     | 827 |
| 7.56.1.53 nppiConvert_32s16s_C1RSfs . . . . . | 828 |
| 7.56.1.54 nppiConvert_32s16u_C1RSfs . . . . . | 828 |
| 7.56.1.55 nppiConvert_32s32f_C1R . . . . .    | 829 |
| 7.56.1.56 nppiConvert_32s32u_C1Rs . . . . .   | 829 |
| 7.56.1.57 nppiConvert_32s8s_AC4R . . . . .    | 829 |
| 7.56.1.58 nppiConvert_32s8s_C1R . . . . .     | 830 |
| 7.56.1.59 nppiConvert_32s8s_C3R . . . . .     | 830 |
| 7.56.1.60 nppiConvert_32s8s_C4R . . . . .     | 830 |
| 7.56.1.61 nppiConvert_32s8u_AC4R . . . . .    | 831 |
| 7.56.1.62 nppiConvert_32s8u_C1R . . . . .     | 831 |
| 7.56.1.63 nppiConvert_32s8u_C3R . . . . .     | 831 |
| 7.56.1.64 nppiConvert_32s8u_C4R . . . . .     | 832 |
| 7.56.1.65 nppiConvert_32u16s_C1RSfs . . . . . | 832 |
| 7.56.1.66 nppiConvert_32u16u_C1RSfs . . . . . | 832 |
| 7.56.1.67 nppiConvert_32u32f_C1R . . . . .    | 833 |
| 7.56.1.68 nppiConvert_32u32s_C1RSfs . . . . . | 833 |
| 7.56.1.69 nppiConvert_32u8s_C1RSfs . . . . .  | 834 |
| 7.56.1.70 nppiConvert_32u8u_C1RSfs . . . . .  | 834 |
| 7.56.1.71 nppiConvert_8s16s_C1R . . . . .     | 834 |
| 7.56.1.72 nppiConvert_8s16u_C1Rs . . . . .    | 835 |

|           |   |     |
|-----------|---|-----|
| 7.56.1.73 | <a href="#">nppiConvert_8s32f_AC4R</a>  | 835 |
| 7.56.1.74 | <a href="#">nppiConvert_8s32f_C1R</a>   | 835 |
| 7.56.1.75 | <a href="#">nppiConvert_8s32f_C3R</a>   | 836 |
| 7.56.1.76 | <a href="#">nppiConvert_8s32f_C4R</a>   | 836 |
| 7.56.1.77 | <a href="#">nppiConvert_8s32s_AC4R</a>  | 837 |
| 7.56.1.78 | <a href="#">nppiConvert_8s32s_C1R</a>   | 837 |
| 7.56.1.79 | <a href="#">nppiConvert_8s32s_C3R</a>   | 837 |
| 7.56.1.80 | <a href="#">nppiConvert_8s32s_C4R</a>   | 838 |
| 7.56.1.81 | <a href="#">nppiConvert_8s32u_C1Rs</a>  | 838 |
| 7.56.1.82 | <a href="#">nppiConvert_8s8u_C1Rs</a>   | 838 |
| 7.56.1.83 | <a href="#">nppiConvert_8u16s_AC4R</a>  | 839 |
| 7.56.1.84 | <a href="#">nppiConvert_8u16s_C1R</a>   | 839 |
| 7.56.1.85 | <a href="#">nppiConvert_8u16s_C3R</a>   | 839 |
| 7.56.1.86 | <a href="#">nppiConvert_8u16s_C4R</a>   | 840 |
| 7.56.1.87 | <a href="#">nppiConvert_8u16u_AC4R</a>  | 840 |
| 7.56.1.88 | <a href="#">nppiConvert_8u16u_C1R</a>   | 840 |
| 7.56.1.89 | <a href="#">nppiConvert_8u16u_C3R</a>   | 841 |
| 7.56.1.90 | <a href="#">nppiConvert_8u16u_C4R</a>   | 841 |
| 7.56.1.91 | <a href="#">nppiConvert_8u32f_AC4R</a>  | 841 |
| 7.56.1.92 | <a href="#">nppiConvert_8u32f_C1R</a>   | 842 |
| 7.56.1.93 | <a href="#">nppiConvert_8u32f_C3R</a>   | 842 |
| 7.56.1.94 | <a href="#">nppiConvert_8u32f_C4R</a>   | 842 |
| 7.56.1.95 | <a href="#">nppiConvert_8u32s_AC4R</a>  | 843 |
| 7.56.1.96 | <a href="#">nppiConvert_8u32s_C1R</a>   | 843 |
| 7.56.1.97 | <a href="#">nppiConvert_8u32s_C3R</a>   | 843 |
| 7.56.1.98 | <a href="#">nppiConvert_8u32s_C4R</a>   | 844 |
| 7.56.1.99 | <a href="#">nppiConvert_8u8s_C1RSfs</a> | 844 |
| 7.57      | <a href="#">Scale</a>                   | 845 |
| 7.57.1    | <a href="#">Function Documentation</a>  | 848 |
| 7.57.1.1  | <a href="#">nppiScale_16s8u_AC4R</a>    | 848 |
| 7.57.1.2  | <a href="#">nppiScale_16s8u_C1R</a>     | 848 |
| 7.57.1.3  | <a href="#">nppiScale_16s8u_C3R</a>     | 848 |
| 7.57.1.4  | <a href="#">nppiScale_16s8u_C4R</a>     | 849 |
| 7.57.1.5  | <a href="#">nppiScale_16u8u_AC4R</a>    | 849 |
| 7.57.1.6  | <a href="#">nppiScale_16u8u_C1R</a>     | 850 |
| 7.57.1.7  | <a href="#">nppiScale_16u8u_C3R</a>     | 850 |

|           |                              |     |
|-----------|------------------------------|-----|
| 7.57.1.8  | nppiScale_16u8u_C4R          | 850 |
| 7.57.1.9  | nppiScale_32f8u_AC4R         | 851 |
| 7.57.1.10 | nppiScale_32f8u_C1R          | 851 |
| 7.57.1.11 | nppiScale_32f8u_C3R          | 851 |
| 7.57.1.12 | nppiScale_32f8u_C4R          | 852 |
| 7.57.1.13 | nppiScale_32s8u_AC4R         | 852 |
| 7.57.1.14 | nppiScale_32s8u_C1R          | 853 |
| 7.57.1.15 | nppiScale_32s8u_C3R          | 853 |
| 7.57.1.16 | nppiScale_32s8u_C4R          | 853 |
| 7.57.1.17 | nppiScale_8u16s_AC4R         | 854 |
| 7.57.1.18 | nppiScale_8u16s_C1R          | 854 |
| 7.57.1.19 | nppiScale_8u16s_C3R          | 854 |
| 7.57.1.20 | nppiScale_8u16s_C4R          | 855 |
| 7.57.1.21 | nppiScale_8u16u_AC4R         | 855 |
| 7.57.1.22 | nppiScale_8u16u_C1R          | 855 |
| 7.57.1.23 | nppiScale_8u16u_C3R          | 856 |
| 7.57.1.24 | nppiScale_8u16u_C4R          | 856 |
| 7.57.1.25 | nppiScale_8u32f_AC4R         | 856 |
| 7.57.1.26 | nppiScale_8u32f_C1R          | 857 |
| 7.57.1.27 | nppiScale_8u32f_C3R          | 857 |
| 7.57.1.28 | nppiScale_8u32f_C4R          | 857 |
| 7.57.1.29 | nppiScale_8u32s_AC4R         | 858 |
| 7.57.1.30 | nppiScale_8u32s_C1R          | 858 |
| 7.57.1.31 | nppiScale_8u32s_C3R          | 859 |
| 7.57.1.32 | nppiScale_8u32s_C4R          | 859 |
| 7.58      | Copy Constant Border         | 860 |
| 7.58.1    | Function Documentation       | 862 |
| 7.58.1.1  | nppiCopyConstBorder_16s_AC4R | 862 |
| 7.58.1.2  | nppiCopyConstBorder_16s_C1R  | 862 |
| 7.58.1.3  | nppiCopyConstBorder_16s_C3R  | 863 |
| 7.58.1.4  | nppiCopyConstBorder_16s_C4R  | 863 |
| 7.58.1.5  | nppiCopyConstBorder_16u_AC4R | 864 |
| 7.58.1.6  | nppiCopyConstBorder_16u_C1R  | 864 |
| 7.58.1.7  | nppiCopyConstBorder_16u_C3R  | 865 |
| 7.58.1.8  | nppiCopyConstBorder_16u_C4R  | 865 |
| 7.58.1.9  | nppiCopyConstBorder_32f_AC4R | 866 |

|           |  |     |
|-----------|--|-----|
| 7.58.1.10 | <a href="#">nppiCopyConstBorder_32f_C1R</a>      | 866 |
| 7.58.1.11 | <a href="#">nppiCopyConstBorder_32f_C3R</a>      | 867 |
| 7.58.1.12 | <a href="#">nppiCopyConstBorder_32f_C4R</a>      | 867 |
| 7.58.1.13 | <a href="#">nppiCopyConstBorder_32s_AC4R</a>     | 868 |
| 7.58.1.14 | <a href="#">nppiCopyConstBorder_32s_C1R</a>      | 868 |
| 7.58.1.15 | <a href="#">nppiCopyConstBorder_32s_C3R</a>      | 869 |
| 7.58.1.16 | <a href="#">nppiCopyConstBorder_32s_C4R</a>      | 869 |
| 7.58.1.17 | <a href="#">nppiCopyConstBorder_8u_AC4R</a>      | 870 |
| 7.58.1.18 | <a href="#">nppiCopyConstBorder_8u_C1R</a>       | 870 |
| 7.58.1.19 | <a href="#">nppiCopyConstBorder_8u_C3R</a>       | 871 |
| 7.58.1.20 | <a href="#">nppiCopyConstBorder_8u_C4R</a>       | 871 |
| 7.59      | <a href="#">Copy Replicate Border</a>            | 873 |
| 7.59.1    | <a href="#">Function Documentation</a>           | 875 |
| 7.59.1.1  | <a href="#">nppiCopyReplicateBorder_16s_AC4R</a> | 875 |
| 7.59.1.2  | <a href="#">nppiCopyReplicateBorder_16s_C1R</a>  | 875 |
| 7.59.1.3  | <a href="#">nppiCopyReplicateBorder_16s_C3R</a>  | 876 |
| 7.59.1.4  | <a href="#">nppiCopyReplicateBorder_16s_C4R</a>  | 876 |
| 7.59.1.5  | <a href="#">nppiCopyReplicateBorder_16u_AC4R</a> | 877 |
| 7.59.1.6  | <a href="#">nppiCopyReplicateBorder_16u_C1R</a>  | 877 |
| 7.59.1.7  | <a href="#">nppiCopyReplicateBorder_16u_C3R</a>  | 878 |
| 7.59.1.8  | <a href="#">nppiCopyReplicateBorder_16u_C4R</a>  | 878 |
| 7.59.1.9  | <a href="#">nppiCopyReplicateBorder_32f_AC4R</a> | 879 |
| 7.59.1.10 | <a href="#">nppiCopyReplicateBorder_32f_C1R</a>  | 879 |
| 7.59.1.11 | <a href="#">nppiCopyReplicateBorder_32f_C3R</a>  | 880 |
| 7.59.1.12 | <a href="#">nppiCopyReplicateBorder_32f_C4R</a>  | 880 |
| 7.59.1.13 | <a href="#">nppiCopyReplicateBorder_32s_AC4R</a> | 881 |
| 7.59.1.14 | <a href="#">nppiCopyReplicateBorder_32s_C1R</a>  | 881 |
| 7.59.1.15 | <a href="#">nppiCopyReplicateBorder_32s_C3R</a>  | 882 |
| 7.59.1.16 | <a href="#">nppiCopyReplicateBorder_32s_C4R</a>  | 882 |
| 7.59.1.17 | <a href="#">nppiCopyReplicateBorder_8u_AC4R</a>  | 883 |
| 7.59.1.18 | <a href="#">nppiCopyReplicateBorder_8u_C1R</a>   | 883 |
| 7.59.1.19 | <a href="#">nppiCopyReplicateBorder_8u_C3R</a>   | 884 |
| 7.59.1.20 | <a href="#">nppiCopyReplicateBorder_8u_C4R</a>   | 884 |
| 7.60      | <a href="#">Copy Wrap Border</a>                 | 885 |
| 7.60.1    | <a href="#">Function Documentation</a>           | 887 |
| 7.60.1.1  | <a href="#">nppiCopyWrapBorder_16s_AC4R</a>      | 887 |

|           |   |     |
|-----------|---|-----|
| 7.60.1.2  | <a href="#">nppiCopyWrapBorder_16s_C1R</a>  | 888 |
| 7.60.1.3  | <a href="#">nppiCopyWrapBorder_16s_C3R</a>  | 888 |
| 7.60.1.4  | <a href="#">nppiCopyWrapBorder_16s_C4R</a>  | 889 |
| 7.60.1.5  | <a href="#">nppiCopyWrapBorder_16u_AC4R</a> | 889 |
| 7.60.1.6  | <a href="#">nppiCopyWrapBorder_16u_C1R</a>  | 890 |
| 7.60.1.7  | <a href="#">nppiCopyWrapBorder_16u_C3R</a>  | 890 |
| 7.60.1.8  | <a href="#">nppiCopyWrapBorder_16u_C4R</a>  | 891 |
| 7.60.1.9  | <a href="#">nppiCopyWrapBorder_32f_AC4R</a> | 891 |
| 7.60.1.10 | <a href="#">nppiCopyWrapBorder_32f_C1R</a>  | 892 |
| 7.60.1.11 | <a href="#">nppiCopyWrapBorder_32f_C3R</a>  | 892 |
| 7.60.1.12 | <a href="#">nppiCopyWrapBorder_32f_C4R</a>  | 893 |
| 7.60.1.13 | <a href="#">nppiCopyWrapBorder_32s_AC4R</a> | 893 |
| 7.60.1.14 | <a href="#">nppiCopyWrapBorder_32s_C1R</a>  | 894 |
| 7.60.1.15 | <a href="#">nppiCopyWrapBorder_32s_C3R</a>  | 894 |
| 7.60.1.16 | <a href="#">nppiCopyWrapBorder_32s_C4R</a>  | 895 |
| 7.60.1.17 | <a href="#">nppiCopyWrapBorder_8u_AC4R</a>  | 895 |
| 7.60.1.18 | <a href="#">nppiCopyWrapBorder_8u_C1R</a>   | 896 |
| 7.60.1.19 | <a href="#">nppiCopyWrapBorder_8u_C3R</a>   | 896 |
| 7.60.1.20 | <a href="#">nppiCopyWrapBorder_8u_C4R</a>   | 897 |
| 7.61      | <a href="#">Copy Sub-Pixel</a>              | 898 |
| 7.61.1    | <a href="#">Function Documentation</a>      | 899 |
| 7.61.1.1  | <a href="#">nppiCopySubpix_16s_AC4R</a>     | 899 |
| 7.61.1.2  | <a href="#">nppiCopySubpix_16s_C1R</a>      | 900 |
| 7.61.1.3  | <a href="#">nppiCopySubpix_16s_C3R</a>      | 900 |
| 7.61.1.4  | <a href="#">nppiCopySubpix_16s_C4R</a>      | 901 |
| 7.61.1.5  | <a href="#">nppiCopySubpix_16u_AC4R</a>     | 901 |
| 7.61.1.6  | <a href="#">nppiCopySubpix_16u_C1R</a>      | 902 |
| 7.61.1.7  | <a href="#">nppiCopySubpix_16u_C3R</a>      | 902 |
| 7.61.1.8  | <a href="#">nppiCopySubpix_16u_C4R</a>      | 902 |
| 7.61.1.9  | <a href="#">nppiCopySubpix_32f_AC4R</a>     | 903 |
| 7.61.1.10 | <a href="#">nppiCopySubpix_32f_C1R</a>      | 903 |
| 7.61.1.11 | <a href="#">nppiCopySubpix_32f_C3R</a>      | 904 |
| 7.61.1.12 | <a href="#">nppiCopySubpix_32f_C4R</a>      | 904 |
| 7.61.1.13 | <a href="#">nppiCopySubpix_32s_AC4R</a>     | 904 |
| 7.61.1.14 | <a href="#">nppiCopySubpix_32s_C1R</a>      | 905 |
| 7.61.1.15 | <a href="#">nppiCopySubpix_32s_C3R</a>      | 905 |

|           |  |     |
|-----------|--|-----|
| 7.61.1.16 | <a href="#">nppiCopySubpix_32s_C4R</a> | 906 |
| 7.61.1.17 | <a href="#">nppiCopySubpix_8u_AC4R</a> | 906 |
| 7.61.1.18 | <a href="#">nppiCopySubpix_8u_C1R</a>  | 907 |
| 7.61.1.19 | <a href="#">nppiCopySubpix_8u_C3R</a>  | 907 |
| 7.61.1.20 | <a href="#">nppiCopySubpix_8u_C4R</a>  | 907 |
| 7.62      | <a href="#">Duplicate Channel</a>      | 909 |
| 7.62.1    | <a href="#">Function Documentation</a> | 910 |
| 7.62.1.1  | <a href="#">nppiDup_16s_C1AC4R</a>     | 910 |
| 7.62.1.2  | <a href="#">nppiDup_16s_C1C3R</a>      | 910 |
| 7.62.1.3  | <a href="#">nppiDup_16s_C1C4R</a>      | 911 |
| 7.62.1.4  | <a href="#">nppiDup_16u_C1AC4R</a>     | 911 |
| 7.62.1.5  | <a href="#">nppiDup_16u_C1C3R</a>      | 912 |
| 7.62.1.6  | <a href="#">nppiDup_16u_C1C4R</a>      | 912 |
| 7.62.1.7  | <a href="#">nppiDup_32f_C1AC4R</a>     | 912 |
| 7.62.1.8  | <a href="#">nppiDup_32f_C1C3R</a>      | 913 |
| 7.62.1.9  | <a href="#">nppiDup_32f_C1C4R</a>      | 913 |
| 7.62.1.10 | <a href="#">nppiDup_32s_C1AC4R</a>     | 913 |
| 7.62.1.11 | <a href="#">nppiDup_32s_C1C3R</a>      | 914 |
| 7.62.1.12 | <a href="#">nppiDup_32s_C1C4R</a>      | 914 |
| 7.62.1.13 | <a href="#">nppiDup_8u_C1AC4R</a>      | 914 |
| 7.62.1.14 | <a href="#">nppiDup_8u_C1C3R</a>       | 915 |
| 7.62.1.15 | <a href="#">nppiDup_8u_C1C4R</a>       | 915 |
| 7.63      | <a href="#">Transpose</a>              | 916 |
| 7.63.1    | <a href="#">Function Documentation</a> | 917 |
| 7.63.1.1  | <a href="#">nppiTranspose_16s_C1R</a>  | 917 |
| 7.63.1.2  | <a href="#">nppiTranspose_16s_C3R</a>  | 917 |
| 7.63.1.3  | <a href="#">nppiTranspose_16s_C4R</a>  | 918 |
| 7.63.1.4  | <a href="#">nppiTranspose_16u_C1R</a>  | 918 |
| 7.63.1.5  | <a href="#">nppiTranspose_16u_C3R</a>  | 918 |
| 7.63.1.6  | <a href="#">nppiTranspose_16u_C4R</a>  | 919 |
| 7.63.1.7  | <a href="#">nppiTranspose_32f_C1R</a>  | 919 |
| 7.63.1.8  | <a href="#">nppiTranspose_32f_C3R</a>  | 919 |
| 7.63.1.9  | <a href="#">nppiTranspose_32f_C4R</a>  | 920 |
| 7.63.1.10 | <a href="#">nppiTranspose_32s_C1R</a>  | 920 |
| 7.63.1.11 | <a href="#">nppiTranspose_32s_C3R</a>  | 921 |
| 7.63.1.12 | <a href="#">nppiTranspose_32s_C4R</a>  | 921 |

|           |  |     |
|-----------|--|-----|
| 7.63.1.13 | <a href="#">nppiTranspose_8u_C1R</a>       | 921 |
| 7.63.1.14 | <a href="#">nppiTranspose_8u_C3R</a>       | 922 |
| 7.63.1.15 | <a href="#">nppiTranspose_8u_C4R</a>       | 922 |
| 7.64      | <a href="#">Swap Channels</a>              | 923 |
| 7.64.1    | <a href="#">Function Documentation</a>     | 926 |
| 7.64.1.1  | <a href="#">nppiSwapChannels_16s_AC4R</a>  | 926 |
| 7.64.1.2  | <a href="#">nppiSwapChannels_16s_C3C4R</a> | 926 |
| 7.64.1.3  | <a href="#">nppiSwapChannels_16s_C3IR</a>  | 927 |
| 7.64.1.4  | <a href="#">nppiSwapChannels_16s_C3R</a>   | 927 |
| 7.64.1.5  | <a href="#">nppiSwapChannels_16s_C4C3R</a> | 927 |
| 7.64.1.6  | <a href="#">nppiSwapChannels_16s_C4IR</a>  | 928 |
| 7.64.1.7  | <a href="#">nppiSwapChannels_16s_C4R</a>   | 928 |
| 7.64.1.8  | <a href="#">nppiSwapChannels_16u_AC4R</a>  | 929 |
| 7.64.1.9  | <a href="#">nppiSwapChannels_16u_C3C4R</a> | 929 |
| 7.64.1.10 | <a href="#">nppiSwapChannels_16u_C3IR</a>  | 930 |
| 7.64.1.11 | <a href="#">nppiSwapChannels_16u_C3R</a>   | 930 |
| 7.64.1.12 | <a href="#">nppiSwapChannels_16u_C4C3R</a> | 930 |
| 7.64.1.13 | <a href="#">nppiSwapChannels_16u_C4IR</a>  | 931 |
| 7.64.1.14 | <a href="#">nppiSwapChannels_16u_C4R</a>   | 931 |
| 7.64.1.15 | <a href="#">nppiSwapChannels_32f_AC4R</a>  | 932 |
| 7.64.1.16 | <a href="#">nppiSwapChannels_32f_C3C4R</a> | 932 |
| 7.64.1.17 | <a href="#">nppiSwapChannels_32f_C3IR</a>  | 933 |
| 7.64.1.18 | <a href="#">nppiSwapChannels_32f_C3R</a>   | 933 |
| 7.64.1.19 | <a href="#">nppiSwapChannels_32f_C4C3R</a> | 933 |
| 7.64.1.20 | <a href="#">nppiSwapChannels_32f_C4IR</a>  | 934 |
| 7.64.1.21 | <a href="#">nppiSwapChannels_32f_C4R</a>   | 934 |
| 7.64.1.22 | <a href="#">nppiSwapChannels_32s_AC4R</a>  | 935 |
| 7.64.1.23 | <a href="#">nppiSwapChannels_32s_C3C4R</a> | 935 |
| 7.64.1.24 | <a href="#">nppiSwapChannels_32s_C3IR</a>  | 936 |
| 7.64.1.25 | <a href="#">nppiSwapChannels_32s_C3R</a>   | 936 |
| 7.64.1.26 | <a href="#">nppiSwapChannels_32s_C4C3R</a> | 936 |
| 7.64.1.27 | <a href="#">nppiSwapChannels_32s_C4IR</a>  | 937 |
| 7.64.1.28 | <a href="#">nppiSwapChannels_32s_C4R</a>   | 937 |
| 7.64.1.29 | <a href="#">nppiSwapChannels_8u_AC4R</a>   | 938 |
| 7.64.1.30 | <a href="#">nppiSwapChannels_8u_C3C4R</a>  | 938 |
| 7.64.1.31 | <a href="#">nppiSwapChannels_8u_C3IR</a>   | 939 |

|           |   |     |
|-----------|---|-----|
| 7.64.1.32 | <a href="#">nppiSwapChannels_8u_C3R</a>     | 939 |
| 7.64.1.33 | <a href="#">nppiSwapChannels_8u_C4C3R</a>   | 939 |
| 7.64.1.34 | <a href="#">nppiSwapChannels_8u_C4IR</a>    | 940 |
| 7.64.1.35 | <a href="#">nppiSwapChannels_8u_C4R</a>     | 940 |
| 7.65      | <a href="#">Filtering Functions</a>         | 941 |
| 7.65.1    | <a href="#">Detailed Description</a>        | 951 |
| 7.65.2    | <a href="#">Function Documentation</a>      | 951 |
| 7.65.2.1  | <a href="#">nppiFilterGauss_16s_AC4R</a>    | 951 |
| 7.65.2.2  | <a href="#">nppiFilterGauss_16s_C1R</a>     | 952 |
| 7.65.2.3  | <a href="#">nppiFilterGauss_16s_C3R</a>     | 952 |
| 7.65.2.4  | <a href="#">nppiFilterGauss_16s_C4R</a>     | 953 |
| 7.65.2.5  | <a href="#">nppiFilterGauss_16u_AC4R</a>    | 953 |
| 7.65.2.6  | <a href="#">nppiFilterGauss_16u_C1R</a>     | 953 |
| 7.65.2.7  | <a href="#">nppiFilterGauss_16u_C3R</a>     | 954 |
| 7.65.2.8  | <a href="#">nppiFilterGauss_16u_C4R</a>     | 954 |
| 7.65.2.9  | <a href="#">nppiFilterGauss_32f_AC4R</a>    | 954 |
| 7.65.2.10 | <a href="#">nppiFilterGauss_32f_C1R</a>     | 955 |
| 7.65.2.11 | <a href="#">nppiFilterGauss_32f_C3R</a>     | 955 |
| 7.65.2.12 | <a href="#">nppiFilterGauss_32f_C4R</a>     | 955 |
| 7.65.2.13 | <a href="#">nppiFilterGauss_8u_AC4R</a>     | 956 |
| 7.65.2.14 | <a href="#">nppiFilterGauss_8u_C1R</a>      | 956 |
| 7.65.2.15 | <a href="#">nppiFilterGauss_8u_C3R</a>      | 956 |
| 7.65.2.16 | <a href="#">nppiFilterGauss_8u_C4R</a>      | 957 |
| 7.65.2.17 | <a href="#">nppiFilterHighPass_16s_AC4R</a> | 957 |
| 7.65.2.18 | <a href="#">nppiFilterHighPass_16s_C1R</a>  | 957 |
| 7.65.2.19 | <a href="#">nppiFilterHighPass_16s_C3R</a>  | 958 |
| 7.65.2.20 | <a href="#">nppiFilterHighPass_16s_C4R</a>  | 958 |
| 7.65.2.21 | <a href="#">nppiFilterHighPass_16u_AC4R</a> | 958 |
| 7.65.2.22 | <a href="#">nppiFilterHighPass_16u_C1R</a>  | 959 |
| 7.65.2.23 | <a href="#">nppiFilterHighPass_16u_C3R</a>  | 959 |
| 7.65.2.24 | <a href="#">nppiFilterHighPass_16u_C4R</a>  | 959 |
| 7.65.2.25 | <a href="#">nppiFilterHighPass_32f_AC4R</a> | 960 |
| 7.65.2.26 | <a href="#">nppiFilterHighPass_32f_C1R</a>  | 960 |
| 7.65.2.27 | <a href="#">nppiFilterHighPass_32f_C3R</a>  | 960 |
| 7.65.2.28 | <a href="#">nppiFilterHighPass_32f_C4R</a>  | 961 |
| 7.65.2.29 | <a href="#">nppiFilterHighPass_8u_AC4R</a>  | 961 |



|           |  |     |
|-----------|--|-----|
| 7.65.2.30 | <a href="#">nppiFilterHighPass_8u_C1R</a>      | 961 |
| 7.65.2.31 | <a href="#">nppiFilterHighPass_8u_C3R</a>      | 962 |
| 7.65.2.32 | <a href="#">nppiFilterHighPass_8u_C4R</a>      | 962 |
| 7.65.2.33 | <a href="#">nppiFilterLaplace_16s_AC4R</a>     | 962 |
| 7.65.2.34 | <a href="#">nppiFilterLaplace_16s_C1R</a>      | 963 |
| 7.65.2.35 | <a href="#">nppiFilterLaplace_16s_C3R</a>      | 963 |
| 7.65.2.36 | <a href="#">nppiFilterLaplace_16s_C4R</a>      | 963 |
| 7.65.2.37 | <a href="#">nppiFilterLaplace_32f_AC4R</a>     | 964 |
| 7.65.2.38 | <a href="#">nppiFilterLaplace_32f_C1R</a>      | 964 |
| 7.65.2.39 | <a href="#">nppiFilterLaplace_32f_C3R</a>      | 964 |
| 7.65.2.40 | <a href="#">nppiFilterLaplace_32f_C4R</a>      | 965 |
| 7.65.2.41 | <a href="#">nppiFilterLaplace_8s16s_C1R</a>    | 965 |
| 7.65.2.42 | <a href="#">nppiFilterLaplace_8u16s_C1R</a>    | 965 |
| 7.65.2.43 | <a href="#">nppiFilterLaplace_8u_AC4R</a>      | 966 |
| 7.65.2.44 | <a href="#">nppiFilterLaplace_8u_C1R</a>       | 966 |
| 7.65.2.45 | <a href="#">nppiFilterLaplace_8u_C3R</a>       | 966 |
| 7.65.2.46 | <a href="#">nppiFilterLaplace_8u_C4R</a>       | 967 |
| 7.65.2.47 | <a href="#">nppiFilterLowPass_16s_AC4R</a>     | 967 |
| 7.65.2.48 | <a href="#">nppiFilterLowPass_16s_C1R</a>      | 967 |
| 7.65.2.49 | <a href="#">nppiFilterLowPass_16s_C3R</a>      | 968 |
| 7.65.2.50 | <a href="#">nppiFilterLowPass_16s_C4R</a>      | 968 |
| 7.65.2.51 | <a href="#">nppiFilterLowPass_16u_AC4R</a>     | 968 |
| 7.65.2.52 | <a href="#">nppiFilterLowPass_16u_C1R</a>      | 969 |
| 7.65.2.53 | <a href="#">nppiFilterLowPass_16u_C3R</a>      | 969 |
| 7.65.2.54 | <a href="#">nppiFilterLowPass_16u_C4R</a>      | 969 |
| 7.65.2.55 | <a href="#">nppiFilterLowPass_32f_AC4R</a>     | 970 |
| 7.65.2.56 | <a href="#">nppiFilterLowPass_32f_C1R</a>      | 970 |
| 7.65.2.57 | <a href="#">nppiFilterLowPass_32f_C3R</a>      | 970 |
| 7.65.2.58 | <a href="#">nppiFilterLowPass_32f_C4R</a>      | 971 |
| 7.65.2.59 | <a href="#">nppiFilterLowPass_8u_AC4R</a>      | 971 |
| 7.65.2.60 | <a href="#">nppiFilterLowPass_8u_C1R</a>       | 971 |
| 7.65.2.61 | <a href="#">nppiFilterLowPass_8u_C3R</a>       | 972 |
| 7.65.2.62 | <a href="#">nppiFilterLowPass_8u_C4R</a>       | 972 |
| 7.65.2.63 | <a href="#">nppiFilterRobertsDown_16s_AC4R</a> | 972 |
| 7.65.2.64 | <a href="#">nppiFilterRobertsDown_16s_C1R</a>  | 973 |
| 7.65.2.65 | <a href="#">nppiFilterRobertsDown_16s_C3R</a>  | 973 |

|  |     |
|--|-----|
| 7.65.2.66 nppiFilterRobertsDown_16s_C4R . . . . .  | 973 |
| 7.65.2.67 nppiFilterRobertsDown_32f_AC4R . . . . . | 974 |
| 7.65.2.68 nppiFilterRobertsDown_32f_C1R . . . . .  | 974 |
| 7.65.2.69 nppiFilterRobertsDown_32f_C3R . . . . .  | 974 |
| 7.65.2.70 nppiFilterRobertsDown_32f_C4R . . . . .  | 975 |
| 7.65.2.71 nppiFilterRobertsDown_8u_AC4R . . . . .  | 975 |
| 7.65.2.72 nppiFilterRobertsDown_8u_C1R . . . . .   | 975 |
| 7.65.2.73 nppiFilterRobertsDown_8u_C3R . . . . .   | 976 |
| 7.65.2.74 nppiFilterRobertsDown_8u_C4R . . . . .   | 976 |
| 7.65.2.75 nppiFilterRobertsUp_16s_AC4R . . . . .   | 976 |
| 7.65.2.76 nppiFilterRobertsUp_16s_C1R . . . . .    | 977 |
| 7.65.2.77 nppiFilterRobertsUp_16s_C3R . . . . .    | 977 |
| 7.65.2.78 nppiFilterRobertsUp_16s_C4R . . . . .    | 977 |
| 7.65.2.79 nppiFilterRobertsUp_32f_AC4R . . . . .   | 978 |
| 7.65.2.80 nppiFilterRobertsUp_32f_C1R . . . . .    | 978 |
| 7.65.2.81 nppiFilterRobertsUp_32f_C3R . . . . .    | 978 |
| 7.65.2.82 nppiFilterRobertsUp_32f_C4R . . . . .    | 979 |
| 7.65.2.83 nppiFilterRobertsUp_8u_AC4R . . . . .    | 979 |
| 7.65.2.84 nppiFilterRobertsUp_8u_C1R . . . . .     | 979 |
| 7.65.2.85 nppiFilterRobertsUp_8u_C3R . . . . .     | 980 |
| 7.65.2.86 nppiFilterRobertsUp_8u_C4R . . . . .     | 980 |
| 7.65.2.87 nppiFilterSharpen_16s_AC4R . . . . .     | 980 |
| 7.65.2.88 nppiFilterSharpen_16s_C1R . . . . .      | 981 |
| 7.65.2.89 nppiFilterSharpen_16s_C3R . . . . .      | 981 |
| 7.65.2.90 nppiFilterSharpen_16s_C4R . . . . .      | 981 |
| 7.65.2.91 nppiFilterSharpen_16u_AC4R . . . . .     | 982 |
| 7.65.2.92 nppiFilterSharpen_16u_C1R . . . . .      | 982 |
| 7.65.2.93 nppiFilterSharpen_16u_C3R . . . . .      | 982 |
| 7.65.2.94 nppiFilterSharpen_16u_C4R . . . . .      | 983 |
| 7.65.2.95 nppiFilterSharpen_32f_AC4R . . . . .     | 983 |
| 7.65.2.96 nppiFilterSharpen_32f_C1R . . . . .      | 983 |
| 7.65.2.97 nppiFilterSharpen_32f_C3R . . . . .      | 984 |
| 7.65.2.98 nppiFilterSharpen_32f_C4R . . . . .      | 984 |
| 7.65.2.99 nppiFilterSharpen_8u_AC4R . . . . .      | 984 |
| 7.65.2.100 nppiFilterSharpen_8u_C1R . . . . .      | 985 |
| 7.65.2.101 nppiFilterSharpen_8u_C3R . . . . .      | 985 |

|            |   |      |
|------------|---|------|
| 7.65.2.102 | nppiFilterSharpen_8u_C4R . . . . .                  | 985  |
| 7.65.2.103 | nppiFilterSobelCrossBorder_32f_C1R . . . . .        | 986  |
| 7.65.2.104 | nppiFilterSobelCrossBorder_8s16s_C1R . . . . .      | 986  |
| 7.65.2.105 | nppiFilterSobelCrossBorder_8u16s_C1R . . . . .      | 987  |
| 7.65.2.106 | nppiFilterSobelVertSecondBorder_32f_C1R . . . . .   | 987  |
| 7.65.2.107 | nppiFilterSobelVertSecondBorder_8s16s_C1R . . . . . | 988  |
| 7.65.2.108 | nppiFilterSobelVertSecondBorder_8u16s_C1R . . . . . | 988  |
| 7.66       | 1D Linear Filter . . . . .                          | 989  |
| 7.66.1     | Function Documentation . . . . .                    | 999  |
| 7.66.1.1   | nppiFilterColumn32f_16s_AC4R . . . . .              | 999  |
| 7.66.1.2   | nppiFilterColumn32f_16s_C1R . . . . .               | 1000 |
| 7.66.1.3   | nppiFilterColumn32f_16s_C3R . . . . .               | 1000 |
| 7.66.1.4   | nppiFilterColumn32f_16s_C4R . . . . .               | 1001 |
| 7.66.1.5   | nppiFilterColumn32f_16u_AC4R . . . . .              | 1001 |
| 7.66.1.6   | nppiFilterColumn32f_16u_C1R . . . . .               | 1002 |
| 7.66.1.7   | nppiFilterColumn32f_16u_C3R . . . . .               | 1002 |
| 7.66.1.8   | nppiFilterColumn32f_16u_C4R . . . . .               | 1003 |
| 7.66.1.9   | nppiFilterColumn32f_8u_AC4R . . . . .               | 1003 |
| 7.66.1.10  | nppiFilterColumn32f_8u_C1R . . . . .                | 1004 |
| 7.66.1.11  | nppiFilterColumn32f_8u_C3R . . . . .                | 1004 |
| 7.66.1.12  | nppiFilterColumn32f_8u_C4R . . . . .                | 1005 |
| 7.66.1.13  | nppiFilterColumn_16s_AC4R . . . . .                 | 1005 |
| 7.66.1.14  | nppiFilterColumn_16s_C1R . . . . .                  | 1006 |
| 7.66.1.15  | nppiFilterColumn_16s_C3R . . . . .                  | 1006 |
| 7.66.1.16  | nppiFilterColumn_16s_C4R . . . . .                  | 1007 |
| 7.66.1.17  | nppiFilterColumn_16u_AC4R . . . . .                 | 1007 |
| 7.66.1.18  | nppiFilterColumn_16u_C1R . . . . .                  | 1008 |
| 7.66.1.19  | nppiFilterColumn_16u_C3R . . . . .                  | 1008 |
| 7.66.1.20  | nppiFilterColumn_16u_C4R . . . . .                  | 1009 |
| 7.66.1.21  | nppiFilterColumn_32f_AC4R . . . . .                 | 1009 |
| 7.66.1.22  | nppiFilterColumn_32f_C1R . . . . .                  | 1010 |
| 7.66.1.23  | nppiFilterColumn_32f_C3R . . . . .                  | 1010 |
| 7.66.1.24  | nppiFilterColumn_32f_C4R . . . . .                  | 1011 |
| 7.66.1.25  | nppiFilterColumn_64f_C1R . . . . .                  | 1011 |
| 7.66.1.26  | nppiFilterColumn_8u_AC4R . . . . .                  | 1012 |
| 7.66.1.27  | nppiFilterColumn_8u_C1R . . . . .                   | 1012 |

|   |      |
|---|------|
| 7.66.1.28 nppiFilterColumn_8u_C3R . . . . .             | 1013 |
| 7.66.1.29 nppiFilterColumn_8u_C4R . . . . .             | 1013 |
| 7.66.1.30 nppiFilterRow32f_16s_AC4R . . . . .           | 1014 |
| 7.66.1.31 nppiFilterRow32f_16s_C1R . . . . .            | 1014 |
| 7.66.1.32 nppiFilterRow32f_16s_C3R . . . . .            | 1015 |
| 7.66.1.33 nppiFilterRow32f_16s_C4R . . . . .            | 1015 |
| 7.66.1.34 nppiFilterRow32f_16u_AC4R . . . . .           | 1016 |
| 7.66.1.35 nppiFilterRow32f_16u_C1R . . . . .            | 1016 |
| 7.66.1.36 nppiFilterRow32f_16u_C3R . . . . .            | 1017 |
| 7.66.1.37 nppiFilterRow32f_16u_C4R . . . . .            | 1017 |
| 7.66.1.38 nppiFilterRow32f_8u_AC4R . . . . .            | 1018 |
| 7.66.1.39 nppiFilterRow32f_8u_C1R . . . . .             | 1018 |
| 7.66.1.40 nppiFilterRow32f_8u_C3R . . . . .             | 1019 |
| 7.66.1.41 nppiFilterRow32f_8u_C4R . . . . .             | 1019 |
| 7.66.1.42 nppiFilterRow_16s_AC4R . . . . .              | 1020 |
| 7.66.1.43 nppiFilterRow_16s_C1R . . . . .               | 1020 |
| 7.66.1.44 nppiFilterRow_16s_C3R . . . . .               | 1021 |
| 7.66.1.45 nppiFilterRow_16s_C4R . . . . .               | 1021 |
| 7.66.1.46 nppiFilterRow_16u_AC4R . . . . .              | 1022 |
| 7.66.1.47 nppiFilterRow_16u_C1R . . . . .               | 1022 |
| 7.66.1.48 nppiFilterRow_16u_C3R . . . . .               | 1023 |
| 7.66.1.49 nppiFilterRow_16u_C4R . . . . .               | 1023 |
| 7.66.1.50 nppiFilterRow_32f_AC4R . . . . .              | 1024 |
| 7.66.1.51 nppiFilterRow_32f_C1R . . . . .               | 1024 |
| 7.66.1.52 nppiFilterRow_32f_C3R . . . . .               | 1025 |
| 7.66.1.53 nppiFilterRow_32f_C4R . . . . .               | 1025 |
| 7.66.1.54 nppiFilterRow_64f_C1R . . . . .               | 1026 |
| 7.66.1.55 nppiFilterRow_8u_AC4R . . . . .               | 1026 |
| 7.66.1.56 nppiFilterRow_8u_C1R . . . . .                | 1027 |
| 7.66.1.57 nppiFilterRow_8u_C3R . . . . .                | 1027 |
| 7.66.1.58 nppiFilterRow_8u_C4R . . . . .                | 1028 |
| 7.66.1.59 nppiFilterSobelCross_32f_C1R . . . . .        | 1028 |
| 7.66.1.60 nppiFilterSobelCross_8s16s_C1R . . . . .      | 1028 |
| 7.66.1.61 nppiFilterSobelCross_8u16s_C1R . . . . .      | 1029 |
| 7.66.1.62 nppiFilterSobelHorizBorder_16s_AC4R . . . . . | 1029 |
| 7.66.1.63 nppiFilterSobelHorizBorder_16s_C1R . . . . .  | 1030 |

|           |  |      |
|-----------|--|------|
| 7.66.1.64 | <a href="#">nppiFilterSobelHorizBorder_16s_C3R</a>         | 1030 |
| 7.66.1.65 | <a href="#">nppiFilterSobelHorizBorder_16s_C4R</a>         | 1030 |
| 7.66.1.66 | <a href="#">nppiFilterSobelHorizBorder_32f_AC4R</a>        | 1031 |
| 7.66.1.67 | <a href="#">nppiFilterSobelHorizBorder_32f_C1R</a>         | 1031 |
| 7.66.1.68 | <a href="#">nppiFilterSobelHorizBorder_32f_C3R</a>         | 1032 |
| 7.66.1.69 | <a href="#">nppiFilterSobelHorizBorder_32f_C4R</a>         | 1032 |
| 7.66.1.70 | <a href="#">nppiFilterSobelHorizBorder_8s16s_C1R</a>       | 1033 |
| 7.66.1.71 | <a href="#">nppiFilterSobelHorizBorder_8u16s_C1R</a>       | 1033 |
| 7.66.1.72 | <a href="#">nppiFilterSobelHorizBorder_8u_AC4R</a>         | 1034 |
| 7.66.1.73 | <a href="#">nppiFilterSobelHorizBorder_8u_C1R</a>          | 1034 |
| 7.66.1.74 | <a href="#">nppiFilterSobelHorizBorder_8u_C3R</a>          | 1034 |
| 7.66.1.75 | <a href="#">nppiFilterSobelHorizBorder_8u_C4R</a>          | 1035 |
| 7.66.1.76 | <a href="#">nppiFilterSobelHorizMaskBorder_32f_C1R</a>     | 1035 |
| 7.66.1.77 | <a href="#">nppiFilterSobelHorizSecondBorder_32f_C1R</a>   | 1036 |
| 7.66.1.78 | <a href="#">nppiFilterSobelHorizSecondBorder_8s16s_C1R</a> | 1036 |
| 7.66.1.79 | <a href="#">nppiFilterSobelHorizSecondBorder_8u16s_C1R</a> | 1037 |
| 7.66.1.80 | <a href="#">nppiFilterSobelVertBorder_16s_AC4R</a>         | 1037 |
| 7.66.1.81 | <a href="#">nppiFilterSobelVertBorder_16s_C1R</a>          | 1038 |
| 7.66.1.82 | <a href="#">nppiFilterSobelVertBorder_16s_C3R</a>          | 1038 |
| 7.66.1.83 | <a href="#">nppiFilterSobelVertBorder_16s_C4R</a>          | 1038 |
| 7.66.1.84 | <a href="#">nppiFilterSobelVertBorder_32f_AC4R</a>         | 1039 |
| 7.66.1.85 | <a href="#">nppiFilterSobelVertBorder_32f_C1R</a>          | 1039 |
| 7.66.1.86 | <a href="#">nppiFilterSobelVertBorder_32f_C3R</a>          | 1040 |
| 7.66.1.87 | <a href="#">nppiFilterSobelVertBorder_32f_C4R</a>          | 1040 |
| 7.66.1.88 | <a href="#">nppiFilterSobelVertBorder_8s16s_C1R</a>        | 1041 |
| 7.66.1.89 | <a href="#">nppiFilterSobelVertBorder_8u16s_C1R</a>        | 1041 |
| 7.66.1.90 | <a href="#">nppiFilterSobelVertBorder_8u_AC4R</a>          | 1042 |
| 7.66.1.91 | <a href="#">nppiFilterSobelVertBorder_8u_C1R</a>           | 1042 |
| 7.66.1.92 | <a href="#">nppiFilterSobelVertBorder_8u_C3R</a>           | 1042 |
| 7.66.1.93 | <a href="#">nppiFilterSobelVertBorder_8u_C4R</a>           | 1043 |
| 7.66.1.94 | <a href="#">nppiFilterSobelVertMaskBorder_32f_C1R</a>      | 1043 |
| 7.66.1.95 | <a href="#">nppiFilterSobelVertSecond_32f_C1R</a>          | 1044 |
| 7.66.1.96 | <a href="#">nppiFilterSobelVertSecond_8s16s_C1R</a>        | 1044 |
| 7.66.1.97 | <a href="#">nppiFilterSobelVertSecond_8u16s_C1R</a>        | 1045 |
| 7.67      | <a href="#">1D Window Sum</a>                              | 1046 |
| 7.67.1    | <a href="#">Function Documentation</a>                     | 1047 |

|           |  |      |
|-----------|--|------|
| 7.67.1.1  | <a href="#">nppiSumWindowColumn_16s32f_C1R</a> | 1047 |
| 7.67.1.2  | <a href="#">nppiSumWindowColumn_16s32f_C3R</a> | 1048 |
| 7.67.1.3  | <a href="#">nppiSumWindowColumn_16s32f_C4R</a> | 1048 |
| 7.67.1.4  | <a href="#">nppiSumWindowColumn_16u32f_C1R</a> | 1049 |
| 7.67.1.5  | <a href="#">nppiSumWindowColumn_16u32f_C3R</a> | 1049 |
| 7.67.1.6  | <a href="#">nppiSumWindowColumn_16u32f_C4R</a> | 1050 |
| 7.67.1.7  | <a href="#">nppiSumWindowColumn_8u32f_C1R</a>  | 1050 |
| 7.67.1.8  | <a href="#">nppiSumWindowColumn_8u32f_C3R</a>  | 1051 |
| 7.67.1.9  | <a href="#">nppiSumWindowColumn_8u32f_C4R</a>  | 1051 |
| 7.67.1.10 | <a href="#">nppiSumWindowRow_16s32f_C1R</a>    | 1052 |
| 7.67.1.11 | <a href="#">nppiSumWindowRow_16s32f_C3R</a>    | 1052 |
| 7.67.1.12 | <a href="#">nppiSumWindowRow_16s32f_C4R</a>    | 1053 |
| 7.67.1.13 | <a href="#">nppiSumWindowRow_16u32f_C1R</a>    | 1053 |
| 7.67.1.14 | <a href="#">nppiSumWindowRow_16u32f_C3R</a>    | 1054 |
| 7.67.1.15 | <a href="#">nppiSumWindowRow_16u32f_C4R</a>    | 1054 |
| 7.67.1.16 | <a href="#">nppiSumWindowRow_8u32f_C1R</a>     | 1055 |
| 7.67.1.17 | <a href="#">nppiSumWindowRow_8u32f_C3R</a>     | 1055 |
| 7.67.1.18 | <a href="#">nppiSumWindowRow_8u32f_C4R</a>     | 1056 |
| 7.68      | <a href="#">Convolution</a>                    | 1057 |
| 7.68.1    | <a href="#">Function Documentation</a>         | 1066 |
| 7.68.1.1  | <a href="#">nppiFilter32f_16s_AC4R</a>         | 1066 |
| 7.68.1.2  | <a href="#">nppiFilter32f_16s_C1R</a>          | 1066 |
| 7.68.1.3  | <a href="#">nppiFilter32f_16s_C3R</a>          | 1067 |
| 7.68.1.4  | <a href="#">nppiFilter32f_16s_C4R</a>          | 1067 |
| 7.68.1.5  | <a href="#">nppiFilter32f_16u_AC4R</a>         | 1068 |
| 7.68.1.6  | <a href="#">nppiFilter32f_16u_C1R</a>          | 1068 |
| 7.68.1.7  | <a href="#">nppiFilter32f_16u_C3R</a>          | 1069 |
| 7.68.1.8  | <a href="#">nppiFilter32f_16u_C4R</a>          | 1069 |
| 7.68.1.9  | <a href="#">nppiFilter32f_32s_AC4R</a>         | 1070 |
| 7.68.1.10 | <a href="#">nppiFilter32f_32s_C1R</a>          | 1070 |
| 7.68.1.11 | <a href="#">nppiFilter32f_32s_C3R</a>          | 1071 |
| 7.68.1.12 | <a href="#">nppiFilter32f_32s_C4R</a>          | 1071 |
| 7.68.1.13 | <a href="#">nppiFilter32f_8s16s_AC4R</a>       | 1072 |
| 7.68.1.14 | <a href="#">nppiFilter32f_8s16s_C1R</a>        | 1072 |
| 7.68.1.15 | <a href="#">nppiFilter32f_8s16s_C3R</a>        | 1073 |
| 7.68.1.16 | <a href="#">nppiFilter32f_8s16s_C4R</a>        | 1073 |

|  |      |
|--|------|
| 7.68.1.17 nppiFilter32f_8s_AC4R . . . . .        | 1074 |
| 7.68.1.18 nppiFilter32f_8s_C1R . . . . .         | 1074 |
| 7.68.1.19 nppiFilter32f_8s_C3R . . . . .         | 1075 |
| 7.68.1.20 nppiFilter32f_8s_C4R . . . . .         | 1075 |
| 7.68.1.21 nppiFilter32f_8u16s_AC4R . . . . .     | 1076 |
| 7.68.1.22 nppiFilter32f_8u16s_C1R . . . . .      | 1076 |
| 7.68.1.23 nppiFilter32f_8u16s_C3R . . . . .      | 1077 |
| 7.68.1.24 nppiFilter32f_8u16s_C4R . . . . .      | 1077 |
| 7.68.1.25 nppiFilter32f_8u_AC4R . . . . .        | 1078 |
| 7.68.1.26 nppiFilter32f_8u_C1R . . . . .         | 1078 |
| 7.68.1.27 nppiFilter32f_8u_C3R . . . . .         | 1079 |
| 7.68.1.28 nppiFilter32f_8u_C4R . . . . .         | 1079 |
| 7.68.1.29 nppiFilter_16s_AC4R . . . . .          | 1080 |
| 7.68.1.30 nppiFilter_16s_C1R . . . . .           | 1080 |
| 7.68.1.31 nppiFilter_16s_C3R . . . . .           | 1081 |
| 7.68.1.32 nppiFilter_16s_C4R . . . . .           | 1081 |
| 7.68.1.33 nppiFilter_16u_AC4R . . . . .          | 1082 |
| 7.68.1.34 nppiFilter_16u_C1R . . . . .           | 1082 |
| 7.68.1.35 nppiFilter_16u_C3R . . . . .           | 1083 |
| 7.68.1.36 nppiFilter_16u_C4R . . . . .           | 1083 |
| 7.68.1.37 nppiFilter_32f_AC4R . . . . .          | 1084 |
| 7.68.1.38 nppiFilter_32f_C1R . . . . .           | 1084 |
| 7.68.1.39 nppiFilter_32f_C3R . . . . .           | 1085 |
| 7.68.1.40 nppiFilter_32f_C4R . . . . .           | 1085 |
| 7.68.1.41 nppiFilter_64f_C1R . . . . .           | 1086 |
| 7.68.1.42 nppiFilter_8u_AC4R . . . . .           | 1086 |
| 7.68.1.43 nppiFilter_8u_C1R . . . . .            | 1087 |
| 7.68.1.44 nppiFilter_8u_C3R . . . . .            | 1087 |
| 7.68.1.45 nppiFilter_8u_C4R . . . . .            | 1088 |
| 7.68.1.46 nppiFilterBorder32f_16s_AC4R . . . . . | 1088 |
| 7.68.1.47 nppiFilterBorder32f_16s_C1R . . . . .  | 1089 |
| 7.68.1.48 nppiFilterBorder32f_16s_C3R . . . . .  | 1089 |
| 7.68.1.49 nppiFilterBorder32f_16s_C4R . . . . .  | 1090 |
| 7.68.1.50 nppiFilterBorder32f_16u_AC4R . . . . . | 1090 |
| 7.68.1.51 nppiFilterBorder32f_16u_C1R . . . . .  | 1091 |
| 7.68.1.52 nppiFilterBorder32f_16u_C3R . . . . .  | 1091 |

|  |      |
|--|------|
| 7.68.1.53 nppiFilterBorder32f_16u_C4R . . . . .    | 1092 |
| 7.68.1.54 nppiFilterBorder32f_32s_AC4R . . . . .   | 1092 |
| 7.68.1.55 nppiFilterBorder32f_32s_C1R . . . . .    | 1093 |
| 7.68.1.56 nppiFilterBorder32f_32s_C3R . . . . .    | 1093 |
| 7.68.1.57 nppiFilterBorder32f_32s_C4R . . . . .    | 1094 |
| 7.68.1.58 nppiFilterBorder32f_8s16s_AC4R . . . . . | 1094 |
| 7.68.1.59 nppiFilterBorder32f_8s16s_C1R . . . . .  | 1095 |
| 7.68.1.60 nppiFilterBorder32f_8s16s_C3R . . . . .  | 1095 |
| 7.68.1.61 nppiFilterBorder32f_8s16s_C4R . . . . .  | 1096 |
| 7.68.1.62 nppiFilterBorder32f_8s_AC4R . . . . .    | 1096 |
| 7.68.1.63 nppiFilterBorder32f_8s_C1R . . . . .     | 1097 |
| 7.68.1.64 nppiFilterBorder32f_8s_C3R . . . . .     | 1097 |
| 7.68.1.65 nppiFilterBorder32f_8s_C4R . . . . .     | 1098 |
| 7.68.1.66 nppiFilterBorder32f_8u16s_AC4R . . . . . | 1098 |
| 7.68.1.67 nppiFilterBorder32f_8u16s_C1R . . . . .  | 1099 |
| 7.68.1.68 nppiFilterBorder32f_8u16s_C3R . . . . .  | 1099 |
| 7.68.1.69 nppiFilterBorder32f_8u16s_C4R . . . . .  | 1100 |
| 7.68.1.70 nppiFilterBorder32f_8u_AC4R . . . . .    | 1100 |
| 7.68.1.71 nppiFilterBorder32f_8u_C1R . . . . .     | 1101 |
| 7.68.1.72 nppiFilterBorder32f_8u_C3R . . . . .     | 1101 |
| 7.68.1.73 nppiFilterBorder32f_8u_C4R . . . . .     | 1102 |
| 7.68.1.74 nppiFilterBorder_16s_AC4R . . . . .      | 1102 |
| 7.68.1.75 nppiFilterBorder_16s_C1R . . . . .       | 1103 |
| 7.68.1.76 nppiFilterBorder_16s_C3R . . . . .       | 1103 |
| 7.68.1.77 nppiFilterBorder_16s_C4R . . . . .       | 1104 |
| 7.68.1.78 nppiFilterBorder_16u_AC4R . . . . .      | 1105 |
| 7.68.1.79 nppiFilterBorder_16u_C1R . . . . .       | 1105 |
| 7.68.1.80 nppiFilterBorder_16u_C3R . . . . .       | 1106 |
| 7.68.1.81 nppiFilterBorder_16u_C4R . . . . .       | 1106 |
| 7.68.1.82 nppiFilterBorder_32f_AC4R . . . . .      | 1107 |
| 7.68.1.83 nppiFilterBorder_32f_C1R . . . . .       | 1108 |
| 7.68.1.84 nppiFilterBorder_32f_C3R . . . . .       | 1108 |
| 7.68.1.85 nppiFilterBorder_32f_C4R . . . . .       | 1109 |
| 7.68.1.86 nppiFilterBorder_8u_AC4R . . . . .       | 1109 |
| 7.68.1.87 nppiFilterBorder_8u_C1R . . . . .        | 1110 |
| 7.68.1.88 nppiFilterBorder_8u_C3R . . . . .        | 1110 |



|           |   |      |
|-----------|---|------|
| 7.68.1.89 | <a href="#">nppiFilterBorder_8u_C4R</a> | 1111 |
| 7.69      | 2D Fixed Linear Filters                 | 1112 |
| 7.69.1    | Function Documentation                  | 1113 |
| 7.69.1.1  | <a href="#">nppiFilterBox_16s_AC4R</a>  | 1113 |
| 7.69.1.2  | <a href="#">nppiFilterBox_16s_C1R</a>   | 1114 |
| 7.69.1.3  | <a href="#">nppiFilterBox_16s_C3R</a>   | 1114 |
| 7.69.1.4  | <a href="#">nppiFilterBox_16s_C4R</a>   | 1114 |
| 7.69.1.5  | <a href="#">nppiFilterBox_16u_AC4R</a>  | 1115 |
| 7.69.1.6  | <a href="#">nppiFilterBox_16u_C1R</a>   | 1115 |
| 7.69.1.7  | <a href="#">nppiFilterBox_16u_C3R</a>   | 1116 |
| 7.69.1.8  | <a href="#">nppiFilterBox_16u_C4R</a>   | 1116 |
| 7.69.1.9  | <a href="#">nppiFilterBox_32f_AC4R</a>  | 1116 |
| 7.69.1.10 | <a href="#">nppiFilterBox_32f_C1R</a>   | 1117 |
| 7.69.1.11 | <a href="#">nppiFilterBox_32f_C3R</a>   | 1117 |
| 7.69.1.12 | <a href="#">nppiFilterBox_32f_C4R</a>   | 1118 |
| 7.69.1.13 | <a href="#">nppiFilterBox_64f_C1R</a>   | 1118 |
| 7.69.1.14 | <a href="#">nppiFilterBox_8u_AC4R</a>   | 1118 |
| 7.69.1.15 | <a href="#">nppiFilterBox_8u_C1R</a>    | 1119 |
| 7.69.1.16 | <a href="#">nppiFilterBox_8u_C3R</a>    | 1119 |
| 7.69.1.17 | <a href="#">nppiFilterBox_8u_C4R</a>    | 1120 |
| 7.70      | Rank Filters                            | 1121 |
| 7.70.1    | Function Documentation                  | 1126 |
| 7.70.1.1  | <a href="#">nppiFilterMax_16s_AC4R</a>  | 1126 |
| 7.70.1.2  | <a href="#">nppiFilterMax_16s_C1R</a>   | 1127 |
| 7.70.1.3  | <a href="#">nppiFilterMax_16s_C3R</a>   | 1127 |
| 7.70.1.4  | <a href="#">nppiFilterMax_16s_C4R</a>   | 1127 |
| 7.70.1.5  | <a href="#">nppiFilterMax_16u_AC4R</a>  | 1128 |
| 7.70.1.6  | <a href="#">nppiFilterMax_16u_C1R</a>   | 1128 |
| 7.70.1.7  | <a href="#">nppiFilterMax_16u_C3R</a>   | 1129 |
| 7.70.1.8  | <a href="#">nppiFilterMax_16u_C4R</a>   | 1129 |
| 7.70.1.9  | <a href="#">nppiFilterMax_32f_AC4R</a>  | 1129 |
| 7.70.1.10 | <a href="#">nppiFilterMax_32f_C1R</a>   | 1130 |
| 7.70.1.11 | <a href="#">nppiFilterMax_32f_C3R</a>   | 1130 |
| 7.70.1.12 | <a href="#">nppiFilterMax_32f_C4R</a>   | 1131 |
| 7.70.1.13 | <a href="#">nppiFilterMax_8u_AC4R</a>   | 1131 |
| 7.70.1.14 | <a href="#">nppiFilterMax_8u_C1R</a>    | 1131 |

|  |      |
|--|------|
| 7.70.1.15 nppiFilterMax_8u_C3R . . . . .                   | 1132 |
| 7.70.1.16 nppiFilterMax_8u_C4R . . . . .                   | 1132 |
| 7.70.1.17 nppiFilterMedian_16s_AC4R . . . . .              | 1133 |
| 7.70.1.18 nppiFilterMedian_16s_C1R . . . . .               | 1133 |
| 7.70.1.19 nppiFilterMedian_16s_C3R . . . . .               | 1133 |
| 7.70.1.20 nppiFilterMedian_16s_C4R . . . . .               | 1134 |
| 7.70.1.21 nppiFilterMedian_16u_AC4R . . . . .              | 1134 |
| 7.70.1.22 nppiFilterMedian_16u_C1R . . . . .               | 1135 |
| 7.70.1.23 nppiFilterMedian_16u_C3R . . . . .               | 1135 |
| 7.70.1.24 nppiFilterMedian_16u_C4R . . . . .               | 1136 |
| 7.70.1.25 nppiFilterMedian_32f_AC4R . . . . .              | 1136 |
| 7.70.1.26 nppiFilterMedian_32f_C1R . . . . .               | 1136 |
| 7.70.1.27 nppiFilterMedian_32f_C3R . . . . .               | 1137 |
| 7.70.1.28 nppiFilterMedian_32f_C4R . . . . .               | 1137 |
| 7.70.1.29 nppiFilterMedian_8u_AC4R . . . . .               | 1138 |
| 7.70.1.30 nppiFilterMedian_8u_C1R . . . . .                | 1138 |
| 7.70.1.31 nppiFilterMedian_8u_C3R . . . . .                | 1139 |
| 7.70.1.32 nppiFilterMedian_8u_C4R . . . . .                | 1139 |
| 7.70.1.33 nppiFilterMedianGetBufferSize_16s_AC4R . . . . . | 1139 |
| 7.70.1.34 nppiFilterMedianGetBufferSize_16s_C1R . . . . .  | 1140 |
| 7.70.1.35 nppiFilterMedianGetBufferSize_16s_C3R . . . . .  | 1140 |
| 7.70.1.36 nppiFilterMedianGetBufferSize_16s_C4R . . . . .  | 1140 |
| 7.70.1.37 nppiFilterMedianGetBufferSize_16u_AC4R . . . . . | 1141 |
| 7.70.1.38 nppiFilterMedianGetBufferSize_16u_C1R . . . . .  | 1141 |
| 7.70.1.39 nppiFilterMedianGetBufferSize_16u_C3R . . . . .  | 1141 |
| 7.70.1.40 nppiFilterMedianGetBufferSize_16u_C4R . . . . .  | 1141 |
| 7.70.1.41 nppiFilterMedianGetBufferSize_32f_AC4R . . . . . | 1142 |
| 7.70.1.42 nppiFilterMedianGetBufferSize_32f_C1R . . . . .  | 1142 |
| 7.70.1.43 nppiFilterMedianGetBufferSize_32f_C3R . . . . .  | 1142 |
| 7.70.1.44 nppiFilterMedianGetBufferSize_32f_C4R . . . . .  | 1143 |
| 7.70.1.45 nppiFilterMedianGetBufferSize_8u_AC4R . . . . .  | 1143 |
| 7.70.1.46 nppiFilterMedianGetBufferSize_8u_C1R . . . . .   | 1143 |
| 7.70.1.47 nppiFilterMedianGetBufferSize_8u_C3R . . . . .   | 1143 |
| 7.70.1.48 nppiFilterMedianGetBufferSize_8u_C4R . . . . .   | 1144 |
| 7.70.1.49 nppiFilterMin_16s_AC4R . . . . .                 | 1144 |
| 7.70.1.50 nppiFilterMin_16s_C1R . . . . .                  | 1145 |

|           |   |      |
|-----------|---|------|
| 7.70.1.51 | <a href="#">nppiFilterMin_16s_C3R</a>           | 1145 |
| 7.70.1.52 | <a href="#">nppiFilterMin_16s_C4R</a>           | 1145 |
| 7.70.1.53 | <a href="#">nppiFilterMin_16u_AC4R</a>          | 1146 |
| 7.70.1.54 | <a href="#">nppiFilterMin_16u_C1R</a>           | 1146 |
| 7.70.1.55 | <a href="#">nppiFilterMin_16u_C3R</a>           | 1147 |
| 7.70.1.56 | <a href="#">nppiFilterMin_16u_C4R</a>           | 1147 |
| 7.70.1.57 | <a href="#">nppiFilterMin_32f_AC4R</a>          | 1147 |
| 7.70.1.58 | <a href="#">nppiFilterMin_32f_C1R</a>           | 1148 |
| 7.70.1.59 | <a href="#">nppiFilterMin_32f_C3R</a>           | 1148 |
| 7.70.1.60 | <a href="#">nppiFilterMin_32f_C4R</a>           | 1149 |
| 7.70.1.61 | <a href="#">nppiFilterMin_8u_AC4R</a>           | 1149 |
| 7.70.1.62 | <a href="#">nppiFilterMin_8u_C1R</a>            | 1149 |
| 7.70.1.63 | <a href="#">nppiFilterMin_8u_C3R</a>            | 1150 |
| 7.70.1.64 | <a href="#">nppiFilterMin_8u_C4R</a>            | 1150 |
| 7.71      | <a href="#">Fixed Filters</a>                   | 1151 |
| 7.71.1    | <a href="#">Detailed Description</a>            | 1158 |
| 7.71.2    | <a href="#">Function Documentation</a>          | 1158 |
| 7.71.2.1  | <a href="#">nppiFilterPrewittHoriz_16s_AC4R</a> | 1158 |
| 7.71.2.2  | <a href="#">nppiFilterPrewittHoriz_16s_C1R</a>  | 1159 |
| 7.71.2.3  | <a href="#">nppiFilterPrewittHoriz_16s_C3R</a>  | 1159 |
| 7.71.2.4  | <a href="#">nppiFilterPrewittHoriz_16s_C4R</a>  | 1159 |
| 7.71.2.5  | <a href="#">nppiFilterPrewittHoriz_32f_AC4R</a> | 1160 |
| 7.71.2.6  | <a href="#">nppiFilterPrewittHoriz_32f_C1R</a>  | 1160 |
| 7.71.2.7  | <a href="#">nppiFilterPrewittHoriz_32f_C3R</a>  | 1160 |
| 7.71.2.8  | <a href="#">nppiFilterPrewittHoriz_32f_C4R</a>  | 1161 |
| 7.71.2.9  | <a href="#">nppiFilterPrewittHoriz_8u_AC4R</a>  | 1161 |
| 7.71.2.10 | <a href="#">nppiFilterPrewittHoriz_8u_C1R</a>   | 1161 |
| 7.71.2.11 | <a href="#">nppiFilterPrewittHoriz_8u_C3R</a>   | 1162 |
| 7.71.2.12 | <a href="#">nppiFilterPrewittHoriz_8u_C4R</a>   | 1162 |
| 7.71.2.13 | <a href="#">nppiFilterPrewittVert_16s_AC4R</a>  | 1162 |
| 7.71.2.14 | <a href="#">nppiFilterPrewittVert_16s_C1R</a>   | 1163 |
| 7.71.2.15 | <a href="#">nppiFilterPrewittVert_16s_C3R</a>   | 1163 |
| 7.71.2.16 | <a href="#">nppiFilterPrewittVert_16s_C4R</a>   | 1163 |
| 7.71.2.17 | <a href="#">nppiFilterPrewittVert_32f_AC4R</a>  | 1164 |
| 7.71.2.18 | <a href="#">nppiFilterPrewittVert_32f_C1R</a>   | 1164 |
| 7.71.2.19 | <a href="#">nppiFilterPrewittVert_32f_C3R</a>   | 1164 |

|           |   |      |
|-----------|---|------|
| 7.71.2.20 | <a href="#">nppiFilterPrewittVert_32f_C4R</a>         | 1165 |
| 7.71.2.21 | <a href="#">nppiFilterPrewittVert_8u_AC4R</a>         | 1165 |
| 7.71.2.22 | <a href="#">nppiFilterPrewittVert_8u_C1R</a>          | 1165 |
| 7.71.2.23 | <a href="#">nppiFilterPrewittVert_8u_C3R</a>          | 1166 |
| 7.71.2.24 | <a href="#">nppiFilterPrewittVert_8u_C4R</a>          | 1166 |
| 7.71.2.25 | <a href="#">nppiFilterScharrHoriz_32f_C1R</a>         | 1166 |
| 7.71.2.26 | <a href="#">nppiFilterScharrHoriz_8s16s_C1R</a>       | 1167 |
| 7.71.2.27 | <a href="#">nppiFilterScharrHoriz_8u16s_C1R</a>       | 1167 |
| 7.71.2.28 | <a href="#">nppiFilterScharrHorizBorder_32f_C1R</a>   | 1167 |
| 7.71.2.29 | <a href="#">nppiFilterScharrHorizBorder_8s16s_C1R</a> | 1168 |
| 7.71.2.30 | <a href="#">nppiFilterScharrHorizBorder_8u16s_C1R</a> | 1168 |
| 7.71.2.31 | <a href="#">nppiFilterScharrVert_32f_C1R</a>          | 1169 |
| 7.71.2.32 | <a href="#">nppiFilterScharrVert_8s16s_C1R</a>        | 1169 |
| 7.71.2.33 | <a href="#">nppiFilterScharrVert_8u16s_C1R</a>        | 1169 |
| 7.71.2.34 | <a href="#">nppiFilterScharrVertBorder_32f_C1R</a>    | 1170 |
| 7.71.2.35 | <a href="#">nppiFilterScharrVertBorder_8s16s_C1R</a>  | 1170 |
| 7.71.2.36 | <a href="#">nppiFilterScharrVertBorder_8u16s_C1R</a>  | 1170 |
| 7.71.2.37 | <a href="#">nppiFilterSobelHoriz_16s_AC4R</a>         | 1171 |
| 7.71.2.38 | <a href="#">nppiFilterSobelHoriz_16s_C1R</a>          | 1171 |
| 7.71.2.39 | <a href="#">nppiFilterSobelHoriz_16s_C3R</a>          | 1172 |
| 7.71.2.40 | <a href="#">nppiFilterSobelHoriz_16s_C4R</a>          | 1172 |
| 7.71.2.41 | <a href="#">nppiFilterSobelHoriz_32f_AC4R</a>         | 1172 |
| 7.71.2.42 | <a href="#">nppiFilterSobelHoriz_32f_C1R</a>          | 1173 |
| 7.71.2.43 | <a href="#">nppiFilterSobelHoriz_32f_C3R</a>          | 1173 |
| 7.71.2.44 | <a href="#">nppiFilterSobelHoriz_32f_C4R</a>          | 1173 |
| 7.71.2.45 | <a href="#">nppiFilterSobelHoriz_8s16s_C1R</a>        | 1174 |
| 7.71.2.46 | <a href="#">nppiFilterSobelHoriz_8u16s_C1R</a>        | 1174 |
| 7.71.2.47 | <a href="#">nppiFilterSobelHoriz_8u_AC4R</a>          | 1174 |
| 7.71.2.48 | <a href="#">nppiFilterSobelHoriz_8u_C1R</a>           | 1175 |
| 7.71.2.49 | <a href="#">nppiFilterSobelHoriz_8u_C3R</a>           | 1175 |
| 7.71.2.50 | <a href="#">nppiFilterSobelHoriz_8u_C4R</a>           | 1175 |
| 7.71.2.51 | <a href="#">nppiFilterSobelHorizMask_32f_C1R</a>      | 1176 |
| 7.71.2.52 | <a href="#">nppiFilterSobelHorizSecond_32f_C1R</a>    | 1176 |
| 7.71.2.53 | <a href="#">nppiFilterSobelHorizSecond_8s16s_C1R</a>  | 1176 |
| 7.71.2.54 | <a href="#">nppiFilterSobelHorizSecond_8u16s_C1R</a>  | 1177 |
| 7.71.2.55 | <a href="#">nppiFilterSobelVert_16s_AC4R</a>          | 1177 |

|           |   |      |
|-----------|---|------|
| 7.71.2.56 | <a href="#">nppiFilterSobelVert_16s_C1R</a>       | 1177 |
| 7.71.2.57 | <a href="#">nppiFilterSobelVert_16s_C3R</a>       | 1178 |
| 7.71.2.58 | <a href="#">nppiFilterSobelVert_16s_C4R</a>       | 1178 |
| 7.71.2.59 | <a href="#">nppiFilterSobelVert_32f_AC4R</a>      | 1179 |
| 7.71.2.60 | <a href="#">nppiFilterSobelVert_32f_C1R</a>       | 1179 |
| 7.71.2.61 | <a href="#">nppiFilterSobelVert_32f_C3R</a>       | 1179 |
| 7.71.2.62 | <a href="#">nppiFilterSobelVert_32f_C4R</a>       | 1180 |
| 7.71.2.63 | <a href="#">nppiFilterSobelVert_8s16s_C1R</a>     | 1180 |
| 7.71.2.64 | <a href="#">nppiFilterSobelVert_8u16s_C1R</a>     | 1180 |
| 7.71.2.65 | <a href="#">nppiFilterSobelVert_8u_AC4R</a>       | 1181 |
| 7.71.2.66 | <a href="#">nppiFilterSobelVert_8u_C1R</a>        | 1181 |
| 7.71.2.67 | <a href="#">nppiFilterSobelVert_8u_C3R</a>        | 1181 |
| 7.71.2.68 | <a href="#">nppiFilterSobelVert_8u_C4R</a>        | 1182 |
| 7.71.2.69 | <a href="#">nppiFilterSobelVertMask_32f_C1R</a>   | 1182 |
| 7.72      | <a href="#">Geometry Transforms</a>               | 1183 |
| 7.72.1    | <a href="#">Detailed Description</a>              | 1183 |
| 7.72.2    | <a href="#">Geometric Transform API Specifics</a> | 1183 |
| 7.72.2.1  | <a href="#">Geometric Transforms and ROIs</a>     | 1183 |
| 7.72.2.2  | <a href="#">Pixel Interpolation</a>               | 1183 |
| 7.73      | <a href="#">ResizeSqrPixel</a>                    | 1185 |
| 7.73.1    | <a href="#">Detailed Description</a>              | 1188 |
| 7.73.2    | <a href="#">Error Codes</a>                       | 1188 |
| 7.73.3    | <a href="#">Function Documentation</a>            | 1189 |
| 7.73.3.1  | <a href="#">nppiGetResizeRect</a>                 | 1189 |
| 7.73.3.2  | <a href="#">nppiResizeSqrPixel_16s_AC4R</a>       | 1189 |
| 7.73.3.3  | <a href="#">nppiResizeSqrPixel_16s_C1R</a>        | 1190 |
| 7.73.3.4  | <a href="#">nppiResizeSqrPixel_16s_C3R</a>        | 1190 |
| 7.73.3.5  | <a href="#">nppiResizeSqrPixel_16s_C4R</a>        | 1191 |
| 7.73.3.6  | <a href="#">nppiResizeSqrPixel_16s_P3R</a>        | 1191 |
| 7.73.3.7  | <a href="#">nppiResizeSqrPixel_16s_P4R</a>        | 1192 |
| 7.73.3.8  | <a href="#">nppiResizeSqrPixel_16u_AC4R</a>       | 1193 |
| 7.73.3.9  | <a href="#">nppiResizeSqrPixel_16u_C1R</a>        | 1193 |
| 7.73.3.10 | <a href="#">nppiResizeSqrPixel_16u_C3R</a>        | 1194 |
| 7.73.3.11 | <a href="#">nppiResizeSqrPixel_16u_C4R</a>        | 1194 |
| 7.73.3.12 | <a href="#">nppiResizeSqrPixel_16u_P3R</a>        | 1195 |
| 7.73.3.13 | <a href="#">nppiResizeSqrPixel_16u_P4R</a>        | 1195 |

|           |   |      |
|-----------|---|------|
| 7.73.3.14 | <a href="#">nppiResizeSqrPixel_32f_AC4R</a> | 1196 |
| 7.73.3.15 | <a href="#">nppiResizeSqrPixel_32f_C1R</a>  | 1196 |
| 7.73.3.16 | <a href="#">nppiResizeSqrPixel_32f_C3R</a>  | 1197 |
| 7.73.3.17 | <a href="#">nppiResizeSqrPixel_32f_C4R</a>  | 1197 |
| 7.73.3.18 | <a href="#">nppiResizeSqrPixel_32f_P3R</a>  | 1198 |
| 7.73.3.19 | <a href="#">nppiResizeSqrPixel_32f_P4R</a>  | 1199 |
| 7.73.3.20 | <a href="#">nppiResizeSqrPixel_64f_AC4R</a> | 1199 |
| 7.73.3.21 | <a href="#">nppiResizeSqrPixel_64f_C1R</a>  | 1200 |
| 7.73.3.22 | <a href="#">nppiResizeSqrPixel_64f_C3R</a>  | 1200 |
| 7.73.3.23 | <a href="#">nppiResizeSqrPixel_64f_C4R</a>  | 1201 |
| 7.73.3.24 | <a href="#">nppiResizeSqrPixel_64f_P3R</a>  | 1201 |
| 7.73.3.25 | <a href="#">nppiResizeSqrPixel_64f_P4R</a>  | 1202 |
| 7.73.3.26 | <a href="#">nppiResizeSqrPixel_8u_AC4R</a>  | 1203 |
| 7.73.3.27 | <a href="#">nppiResizeSqrPixel_8u_C1R</a>   | 1203 |
| 7.73.3.28 | <a href="#">nppiResizeSqrPixel_8u_C3R</a>   | 1204 |
| 7.73.3.29 | <a href="#">nppiResizeSqrPixel_8u_C4R</a>   | 1204 |
| 7.73.3.30 | <a href="#">nppiResizeSqrPixel_8u_P3R</a>   | 1205 |
| 7.73.3.31 | <a href="#">nppiResizeSqrPixel_8u_P4R</a>   | 1205 |
| 7.74      | <a href="#">Resize</a>                      | 1207 |
| 7.74.1    | <a href="#">Detailed Description</a>        | 1208 |
| 7.74.2    | <a href="#">Error Codes</a>                 | 1209 |
| 7.74.3    | <a href="#">Function Documentation</a>      | 1209 |
| 7.74.3.1  | <a href="#">nppiResize_16u_AC4R</a>         | 1209 |
| 7.74.3.2  | <a href="#">nppiResize_16u_C1R</a>          | 1210 |
| 7.74.3.3  | <a href="#">nppiResize_16u_C3R</a>          | 1210 |
| 7.74.3.4  | <a href="#">nppiResize_16u_C4R</a>          | 1211 |
| 7.74.3.5  | <a href="#">nppiResize_16u_P3R</a>          | 1211 |
| 7.74.3.6  | <a href="#">nppiResize_16u_P4R</a>          | 1212 |
| 7.74.3.7  | <a href="#">nppiResize_32f_AC4R</a>         | 1212 |
| 7.74.3.8  | <a href="#">nppiResize_32f_C1R</a>          | 1213 |
| 7.74.3.9  | <a href="#">nppiResize_32f_C3R</a>          | 1213 |
| 7.74.3.10 | <a href="#">nppiResize_32f_C4R</a>          | 1214 |
| 7.74.3.11 | <a href="#">nppiResize_32f_P3R</a>          | 1214 |
| 7.74.3.12 | <a href="#">nppiResize_32f_P4R</a>          | 1215 |
| 7.74.3.13 | <a href="#">nppiResize_8u_AC4R</a>          | 1215 |
| 7.74.3.14 | <a href="#">nppiResize_8u_C1R</a>           | 1216 |

|           |  |      |
|-----------|--|------|
| 7.74.3.15 | <a href="#">nppiResize_8u_C3R</a>      | 1216 |
| 7.74.3.16 | <a href="#">nppiResize_8u_C4R</a>      | 1217 |
| 7.74.3.17 | <a href="#">nppiResize_8u_P3R</a>      | 1217 |
| 7.74.3.18 | <a href="#">nppiResize_8u_P4R</a>      | 1218 |
| 7.75      | <a href="#">Remap</a>                  | 1219 |
| 7.75.1    | <a href="#">Detailed Description</a>   | 1222 |
| 7.75.2    | <a href="#">Error Codes</a>            | 1222 |
| 7.75.3    | <a href="#">Function Documentation</a> | 1222 |
| 7.75.3.1  | <a href="#">nppiRemap_16s_AC4R</a>     | 1222 |
| 7.75.3.2  | <a href="#">nppiRemap_16s_C1R</a>      | 1223 |
| 7.75.3.3  | <a href="#">nppiRemap_16s_C3R</a>      | 1224 |
| 7.75.3.4  | <a href="#">nppiRemap_16s_C4R</a>      | 1224 |
| 7.75.3.5  | <a href="#">nppiRemap_16s_P3R</a>      | 1225 |
| 7.75.3.6  | <a href="#">nppiRemap_16s_P4R</a>      | 1225 |
| 7.75.3.7  | <a href="#">nppiRemap_16u_AC4R</a>     | 1226 |
| 7.75.3.8  | <a href="#">nppiRemap_16u_C1R</a>      | 1227 |
| 7.75.3.9  | <a href="#">nppiRemap_16u_C3R</a>      | 1227 |
| 7.75.3.10 | <a href="#">nppiRemap_16u_C4R</a>      | 1228 |
| 7.75.3.11 | <a href="#">nppiRemap_16u_P3R</a>      | 1228 |
| 7.75.3.12 | <a href="#">nppiRemap_16u_P4R</a>      | 1229 |
| 7.75.3.13 | <a href="#">nppiRemap_32f_AC4R</a>     | 1230 |
| 7.75.3.14 | <a href="#">nppiRemap_32f_C1R</a>      | 1230 |
| 7.75.3.15 | <a href="#">nppiRemap_32f_C3R</a>      | 1231 |
| 7.75.3.16 | <a href="#">nppiRemap_32f_C4R</a>      | 1231 |
| 7.75.3.17 | <a href="#">nppiRemap_32f_P3R</a>      | 1232 |
| 7.75.3.18 | <a href="#">nppiRemap_32f_P4R</a>      | 1233 |
| 7.75.3.19 | <a href="#">nppiRemap_64f_AC4R</a>     | 1233 |
| 7.75.3.20 | <a href="#">nppiRemap_64f_C1R</a>      | 1234 |
| 7.75.3.21 | <a href="#">nppiRemap_64f_C3R</a>      | 1234 |
| 7.75.3.22 | <a href="#">nppiRemap_64f_C4R</a>      | 1235 |
| 7.75.3.23 | <a href="#">nppiRemap_64f_P3R</a>      | 1236 |
| 7.75.3.24 | <a href="#">nppiRemap_64f_P4R</a>      | 1236 |
| 7.75.3.25 | <a href="#">nppiRemap_8u_AC4R</a>      | 1237 |
| 7.75.3.26 | <a href="#">nppiRemap_8u_C1R</a>       | 1237 |
| 7.75.3.27 | <a href="#">nppiRemap_8u_C3R</a>       | 1238 |
| 7.75.3.28 | <a href="#">nppiRemap_8u_C4R</a>       | 1239 |

|           |  |      |
|-----------|--|------|
| 7.75.3.29 | <a href="#">nppiRemap_8u_P3R</a>       | 1239 |
| 7.75.3.30 | <a href="#">nppiRemap_8u_P4R</a>       | 1240 |
| 7.76      | <a href="#">Rotate</a>                 | 1241 |
| 7.76.1    | <a href="#">Detailed Description</a>   | 1242 |
| 7.76.2    | <a href="#">Rotate Error Codes</a>     | 1242 |
| 7.76.3    | <a href="#">Function Documentation</a> | 1242 |
| 7.76.3.1  | <a href="#">nppiGetRotateBound</a>     | 1242 |
| 7.76.3.2  | <a href="#">nppiGetRotateQuad</a>      | 1243 |
| 7.76.3.3  | <a href="#">nppiRotate_16u_AC4R</a>    | 1243 |
| 7.76.3.4  | <a href="#">nppiRotate_16u_C1R</a>     | 1244 |
| 7.76.3.5  | <a href="#">nppiRotate_16u_C3R</a>     | 1244 |
| 7.76.3.6  | <a href="#">nppiRotate_16u_C4R</a>     | 1245 |
| 7.76.3.7  | <a href="#">nppiRotate_32f_AC4R</a>    | 1245 |
| 7.76.3.8  | <a href="#">nppiRotate_32f_C1R</a>     | 1246 |
| 7.76.3.9  | <a href="#">nppiRotate_32f_C3R</a>     | 1246 |
| 7.76.3.10 | <a href="#">nppiRotate_32f_C4R</a>     | 1247 |
| 7.76.3.11 | <a href="#">nppiRotate_8u_AC4R</a>     | 1247 |
| 7.76.3.12 | <a href="#">nppiRotate_8u_C1R</a>      | 1248 |
| 7.76.3.13 | <a href="#">nppiRotate_8u_C3R</a>      | 1248 |
| 7.76.3.14 | <a href="#">nppiRotate_8u_C4R</a>      | 1249 |
| 7.77      | <a href="#">Mirror</a>                 | 1250 |
| 7.77.1    | <a href="#">Detailed Description</a>   | 1253 |
| 7.77.2    | <a href="#">Mirror Error Codes</a>     | 1253 |
| 7.77.3    | <a href="#">Function Documentation</a> | 1253 |
| 7.77.3.1  | <a href="#">nppiMirror_16s_AC4IR</a>   | 1253 |
| 7.77.3.2  | <a href="#">nppiMirror_16s_AC4R</a>    | 1253 |
| 7.77.3.3  | <a href="#">nppiMirror_16s_C1IR</a>    | 1254 |
| 7.77.3.4  | <a href="#">nppiMirror_16s_C1R</a>     | 1254 |
| 7.77.3.5  | <a href="#">nppiMirror_16s_C3IR</a>    | 1254 |
| 7.77.3.6  | <a href="#">nppiMirror_16s_C3R</a>     | 1255 |
| 7.77.3.7  | <a href="#">nppiMirror_16s_C4IR</a>    | 1255 |
| 7.77.3.8  | <a href="#">nppiMirror_16s_C4R</a>     | 1255 |
| 7.77.3.9  | <a href="#">nppiMirror_16u_AC4IR</a>   | 1256 |
| 7.77.3.10 | <a href="#">nppiMirror_16u_AC4R</a>    | 1256 |
| 7.77.3.11 | <a href="#">nppiMirror_16u_C1IR</a>    | 1257 |
| 7.77.3.12 | <a href="#">nppiMirror_16u_C1R</a>     | 1257 |



|           |  |      |
|-----------|--|------|
| 7.77.3.13 | <a href="#">nppiMirror_16u_C3IR</a>          | 1257 |
| 7.77.3.14 | <a href="#">nppiMirror_16u_C3R</a>           | 1258 |
| 7.77.3.15 | <a href="#">nppiMirror_16u_C4IR</a>          | 1258 |
| 7.77.3.16 | <a href="#">nppiMirror_16u_C4R</a>           | 1258 |
| 7.77.3.17 | <a href="#">nppiMirror_32f_AC4IR</a>         | 1259 |
| 7.77.3.18 | <a href="#">nppiMirror_32f_AC4R</a>          | 1259 |
| 7.77.3.19 | <a href="#">nppiMirror_32f_C1IR</a>          | 1259 |
| 7.77.3.20 | <a href="#">nppiMirror_32f_C1R</a>           | 1260 |
| 7.77.3.21 | <a href="#">nppiMirror_32f_C3IR</a>          | 1260 |
| 7.77.3.22 | <a href="#">nppiMirror_32f_C3R</a>           | 1260 |
| 7.77.3.23 | <a href="#">nppiMirror_32f_C4IR</a>          | 1261 |
| 7.77.3.24 | <a href="#">nppiMirror_32f_C4R</a>           | 1261 |
| 7.77.3.25 | <a href="#">nppiMirror_32s_AC4IR</a>         | 1261 |
| 7.77.3.26 | <a href="#">nppiMirror_32s_AC4R</a>          | 1262 |
| 7.77.3.27 | <a href="#">nppiMirror_32s_C1IR</a>          | 1262 |
| 7.77.3.28 | <a href="#">nppiMirror_32s_C1R</a>           | 1262 |
| 7.77.3.29 | <a href="#">nppiMirror_32s_C3IR</a>          | 1263 |
| 7.77.3.30 | <a href="#">nppiMirror_32s_C3R</a>           | 1263 |
| 7.77.3.31 | <a href="#">nppiMirror_32s_C4IR</a>          | 1263 |
| 7.77.3.32 | <a href="#">nppiMirror_32s_C4R</a>           | 1264 |
| 7.77.3.33 | <a href="#">nppiMirror_8u_AC4IR</a>          | 1264 |
| 7.77.3.34 | <a href="#">nppiMirror_8u_AC4R</a>           | 1264 |
| 7.77.3.35 | <a href="#">nppiMirror_8u_C1IR</a>           | 1265 |
| 7.77.3.36 | <a href="#">nppiMirror_8u_C1R</a>            | 1265 |
| 7.77.3.37 | <a href="#">nppiMirror_8u_C3IR</a>           | 1265 |
| 7.77.3.38 | <a href="#">nppiMirror_8u_C3R</a>            | 1266 |
| 7.77.3.39 | <a href="#">nppiMirror_8u_C4IR</a>           | 1266 |
| 7.77.3.40 | <a href="#">nppiMirror_8u_C4R</a>            | 1266 |
| 7.78      | <a href="#">Affine Transforms</a>            | 1267 |
| 7.78.1    | <a href="#">Detailed Description</a>         | 1276 |
| 7.78.2    | <a href="#">Affine Transform Error Codes</a> | 1276 |
| 7.78.3    | <a href="#">Function Documentation</a>       | 1276 |
| 7.78.3.1  | <a href="#">nppiGetAffineBound</a>           | 1276 |
| 7.78.3.2  | <a href="#">nppiGetAffineQuad</a>            | 1276 |
| 7.78.3.3  | <a href="#">nppiGetAffineTransform</a>       | 1277 |
| 7.78.3.4  | <a href="#">nppiWarpAffine_16u_AC4R</a>      | 1278 |

|           |   |      |
|-----------|---|------|
| 7.78.3.5  | <a href="#">nppiWarpAffine_16u_C1R</a>      | 1278 |
| 7.78.3.6  | <a href="#">nppiWarpAffine_16u_C3R</a>      | 1279 |
| 7.78.3.7  | <a href="#">nppiWarpAffine_16u_C4R</a>      | 1279 |
| 7.78.3.8  | <a href="#">nppiWarpAffine_16u_P3R</a>      | 1280 |
| 7.78.3.9  | <a href="#">nppiWarpAffine_16u_P4R</a>      | 1280 |
| 7.78.3.10 | <a href="#">nppiWarpAffine_32f_AC4R</a>     | 1281 |
| 7.78.3.11 | <a href="#">nppiWarpAffine_32f_C1R</a>      | 1281 |
| 7.78.3.12 | <a href="#">nppiWarpAffine_32f_C3R</a>      | 1282 |
| 7.78.3.13 | <a href="#">nppiWarpAffine_32f_C4R</a>      | 1282 |
| 7.78.3.14 | <a href="#">nppiWarpAffine_32f_P3R</a>      | 1283 |
| 7.78.3.15 | <a href="#">nppiWarpAffine_32f_P4R</a>      | 1283 |
| 7.78.3.16 | <a href="#">nppiWarpAffine_32s_AC4R</a>     | 1284 |
| 7.78.3.17 | <a href="#">nppiWarpAffine_32s_C1R</a>      | 1284 |
| 7.78.3.18 | <a href="#">nppiWarpAffine_32s_C3R</a>      | 1285 |
| 7.78.3.19 | <a href="#">nppiWarpAffine_32s_C4R</a>      | 1285 |
| 7.78.3.20 | <a href="#">nppiWarpAffine_32s_P3R</a>      | 1286 |
| 7.78.3.21 | <a href="#">nppiWarpAffine_32s_P4R</a>      | 1286 |
| 7.78.3.22 | <a href="#">nppiWarpAffine_64f_AC4R</a>     | 1287 |
| 7.78.3.23 | <a href="#">nppiWarpAffine_64f_C1R</a>      | 1287 |
| 7.78.3.24 | <a href="#">nppiWarpAffine_64f_C3R</a>      | 1288 |
| 7.78.3.25 | <a href="#">nppiWarpAffine_64f_C4R</a>      | 1288 |
| 7.78.3.26 | <a href="#">nppiWarpAffine_64f_P3R</a>      | 1289 |
| 7.78.3.27 | <a href="#">nppiWarpAffine_64f_P4R</a>      | 1289 |
| 7.78.3.28 | <a href="#">nppiWarpAffine_8u_AC4R</a>      | 1290 |
| 7.78.3.29 | <a href="#">nppiWarpAffine_8u_C1R</a>       | 1290 |
| 7.78.3.30 | <a href="#">nppiWarpAffine_8u_C3R</a>       | 1291 |
| 7.78.3.31 | <a href="#">nppiWarpAffine_8u_C4R</a>       | 1291 |
| 7.78.3.32 | <a href="#">nppiWarpAffine_8u_P3R</a>       | 1292 |
| 7.78.3.33 | <a href="#">nppiWarpAffine_8u_P4R</a>       | 1292 |
| 7.78.3.34 | <a href="#">nppiWarpAffineBack_16u_AC4R</a> | 1293 |
| 7.78.3.35 | <a href="#">nppiWarpAffineBack_16u_C1R</a>  | 1293 |
| 7.78.3.36 | <a href="#">nppiWarpAffineBack_16u_C3R</a>  | 1294 |
| 7.78.3.37 | <a href="#">nppiWarpAffineBack_16u_C4R</a>  | 1294 |
| 7.78.3.38 | <a href="#">nppiWarpAffineBack_16u_P3R</a>  | 1295 |
| 7.78.3.39 | <a href="#">nppiWarpAffineBack_16u_P4R</a>  | 1295 |
| 7.78.3.40 | <a href="#">nppiWarpAffineBack_32f_AC4R</a> | 1296 |

|   |      |
|---|------|
| 7.78.3.41 nppiWarpAffineBack_32f_C1R . . . . .  | 1296 |
| 7.78.3.42 nppiWarpAffineBack_32f_C3R . . . . .  | 1297 |
| 7.78.3.43 nppiWarpAffineBack_32f_C4R . . . . .  | 1297 |
| 7.78.3.44 nppiWarpAffineBack_32f_P3R . . . . .  | 1298 |
| 7.78.3.45 nppiWarpAffineBack_32f_P4R . . . . .  | 1298 |
| 7.78.3.46 nppiWarpAffineBack_32s_AC4R . . . . . | 1299 |
| 7.78.3.47 nppiWarpAffineBack_32s_C1R . . . . .  | 1299 |
| 7.78.3.48 nppiWarpAffineBack_32s_C3R . . . . .  | 1300 |
| 7.78.3.49 nppiWarpAffineBack_32s_C4R . . . . .  | 1300 |
| 7.78.3.50 nppiWarpAffineBack_32s_P3R . . . . .  | 1301 |
| 7.78.3.51 nppiWarpAffineBack_32s_P4R . . . . .  | 1301 |
| 7.78.3.52 nppiWarpAffineBack_8u_AC4R . . . . .  | 1302 |
| 7.78.3.53 nppiWarpAffineBack_8u_C1R . . . . .   | 1302 |
| 7.78.3.54 nppiWarpAffineBack_8u_C3R . . . . .   | 1303 |
| 7.78.3.55 nppiWarpAffineBack_8u_C4R . . . . .   | 1303 |
| 7.78.3.56 nppiWarpAffineBack_8u_P3R . . . . .   | 1304 |
| 7.78.3.57 nppiWarpAffineBack_8u_P4R . . . . .   | 1304 |
| 7.78.3.58 nppiWarpAffineQuad_16u_AC4R . . . . . | 1305 |
| 7.78.3.59 nppiWarpAffineQuad_16u_C1R . . . . .  | 1305 |
| 7.78.3.60 nppiWarpAffineQuad_16u_C3R . . . . .  | 1306 |
| 7.78.3.61 nppiWarpAffineQuad_16u_C4R . . . . .  | 1306 |
| 7.78.3.62 nppiWarpAffineQuad_16u_P3R . . . . .  | 1307 |
| 7.78.3.63 nppiWarpAffineQuad_16u_P4R . . . . .  | 1307 |
| 7.78.3.64 nppiWarpAffineQuad_32f_AC4R . . . . . | 1308 |
| 7.78.3.65 nppiWarpAffineQuad_32f_C1R . . . . .  | 1308 |
| 7.78.3.66 nppiWarpAffineQuad_32f_C3R . . . . .  | 1309 |
| 7.78.3.67 nppiWarpAffineQuad_32f_C4R . . . . .  | 1309 |
| 7.78.3.68 nppiWarpAffineQuad_32f_P3R . . . . .  | 1310 |
| 7.78.3.69 nppiWarpAffineQuad_32f_P4R . . . . .  | 1310 |
| 7.78.3.70 nppiWarpAffineQuad_32s_AC4R . . . . . | 1311 |
| 7.78.3.71 nppiWarpAffineQuad_32s_C1R . . . . .  | 1311 |
| 7.78.3.72 nppiWarpAffineQuad_32s_C3R . . . . .  | 1312 |
| 7.78.3.73 nppiWarpAffineQuad_32s_C4R . . . . .  | 1312 |
| 7.78.3.74 nppiWarpAffineQuad_32s_P3R . . . . .  | 1313 |
| 7.78.3.75 nppiWarpAffineQuad_32s_P4R . . . . .  | 1313 |
| 7.78.3.76 nppiWarpAffineQuad_8u_AC4R . . . . .  | 1314 |

|           |   |      |
|-----------|---|------|
| 7.78.3.77 | <a href="#">nppiWarpAffineQuad_8u_C1R</a>         | 1314 |
| 7.78.3.78 | <a href="#">nppiWarpAffineQuad_8u_C3R</a>         | 1315 |
| 7.78.3.79 | <a href="#">nppiWarpAffineQuad_8u_C4R</a>         | 1315 |
| 7.78.3.80 | <a href="#">nppiWarpAffineQuad_8u_P3R</a>         | 1316 |
| 7.78.3.81 | <a href="#">nppiWarpAffineQuad_8u_P4R</a>         | 1316 |
| 7.79      | <a href="#">Perspective Transform</a>             | 1317 |
| 7.79.1    | <a href="#">Detailed Description</a>              | 1325 |
| 7.79.2    | <a href="#">Perspective Transform Error Codes</a> | 1325 |
| 7.79.3    | <a href="#">Function Documentation</a>            | 1325 |
| 7.79.3.1  | <a href="#">nppiGetPerspectiveBound</a>           | 1325 |
| 7.79.3.2  | <a href="#">nppiGetPerspectiveQuad</a>            | 1326 |
| 7.79.3.3  | <a href="#">nppiGetPerspectiveTransform</a>       | 1326 |
| 7.79.3.4  | <a href="#">nppiWarpPerspective_16u_AC4R</a>      | 1326 |
| 7.79.3.5  | <a href="#">nppiWarpPerspective_16u_C1R</a>       | 1327 |
| 7.79.3.6  | <a href="#">nppiWarpPerspective_16u_C3R</a>       | 1327 |
| 7.79.3.7  | <a href="#">nppiWarpPerspective_16u_C4R</a>       | 1328 |
| 7.79.3.8  | <a href="#">nppiWarpPerspective_16u_P3R</a>       | 1328 |
| 7.79.3.9  | <a href="#">nppiWarpPerspective_16u_P4R</a>       | 1329 |
| 7.79.3.10 | <a href="#">nppiWarpPerspective_32f_AC4R</a>      | 1329 |
| 7.79.3.11 | <a href="#">nppiWarpPerspective_32f_C1R</a>       | 1330 |
| 7.79.3.12 | <a href="#">nppiWarpPerspective_32f_C3R</a>       | 1330 |
| 7.79.3.13 | <a href="#">nppiWarpPerspective_32f_C4R</a>       | 1331 |
| 7.79.3.14 | <a href="#">nppiWarpPerspective_32f_P3R</a>       | 1331 |
| 7.79.3.15 | <a href="#">nppiWarpPerspective_32f_P4R</a>       | 1332 |
| 7.79.3.16 | <a href="#">nppiWarpPerspective_32s_AC4R</a>      | 1332 |
| 7.79.3.17 | <a href="#">nppiWarpPerspective_32s_C1R</a>       | 1333 |
| 7.79.3.18 | <a href="#">nppiWarpPerspective_32s_C3R</a>       | 1333 |
| 7.79.3.19 | <a href="#">nppiWarpPerspective_32s_C4R</a>       | 1334 |
| 7.79.3.20 | <a href="#">nppiWarpPerspective_32s_P3R</a>       | 1334 |
| 7.79.3.21 | <a href="#">nppiWarpPerspective_32s_P4R</a>       | 1335 |
| 7.79.3.22 | <a href="#">nppiWarpPerspective_8u_AC4R</a>       | 1335 |
| 7.79.3.23 | <a href="#">nppiWarpPerspective_8u_C1R</a>        | 1336 |
| 7.79.3.24 | <a href="#">nppiWarpPerspective_8u_C3R</a>        | 1336 |
| 7.79.3.25 | <a href="#">nppiWarpPerspective_8u_C4R</a>        | 1337 |
| 7.79.3.26 | <a href="#">nppiWarpPerspective_8u_P3R</a>        | 1337 |
| 7.79.3.27 | <a href="#">nppiWarpPerspective_8u_P4R</a>        | 1338 |

|  |      |
|--|------|
| 7.79.3.28 nppiWarpPerspectiveBack_16u_AC4R . . . . . | 1338 |
| 7.79.3.29 nppiWarpPerspectiveBack_16u_C1R . . . . .  | 1339 |
| 7.79.3.30 nppiWarpPerspectiveBack_16u_C3R . . . . .  | 1339 |
| 7.79.3.31 nppiWarpPerspectiveBack_16u_C4R . . . . .  | 1340 |
| 7.79.3.32 nppiWarpPerspectiveBack_16u_P3R . . . . .  | 1340 |
| 7.79.3.33 nppiWarpPerspectiveBack_16u_P4R . . . . .  | 1341 |
| 7.79.3.34 nppiWarpPerspectiveBack_32f_AC4R . . . . . | 1341 |
| 7.79.3.35 nppiWarpPerspectiveBack_32f_C1R . . . . .  | 1342 |
| 7.79.3.36 nppiWarpPerspectiveBack_32f_C3R . . . . .  | 1342 |
| 7.79.3.37 nppiWarpPerspectiveBack_32f_C4R . . . . .  | 1343 |
| 7.79.3.38 nppiWarpPerspectiveBack_32f_P3R . . . . .  | 1343 |
| 7.79.3.39 nppiWarpPerspectiveBack_32f_P4R . . . . .  | 1344 |
| 7.79.3.40 nppiWarpPerspectiveBack_32s_AC4R . . . . . | 1344 |
| 7.79.3.41 nppiWarpPerspectiveBack_32s_C1R . . . . .  | 1345 |
| 7.79.3.42 nppiWarpPerspectiveBack_32s_C3R . . . . .  | 1345 |
| 7.79.3.43 nppiWarpPerspectiveBack_32s_C4R . . . . .  | 1346 |
| 7.79.3.44 nppiWarpPerspectiveBack_32s_P3R . . . . .  | 1346 |
| 7.79.3.45 nppiWarpPerspectiveBack_32s_P4R . . . . .  | 1347 |
| 7.79.3.46 nppiWarpPerspectiveBack_8u_AC4R . . . . .  | 1347 |
| 7.79.3.47 nppiWarpPerspectiveBack_8u_C1R . . . . .   | 1348 |
| 7.79.3.48 nppiWarpPerspectiveBack_8u_C3R . . . . .   | 1348 |
| 7.79.3.49 nppiWarpPerspectiveBack_8u_C4R . . . . .   | 1349 |
| 7.79.3.50 nppiWarpPerspectiveBack_8u_P3R . . . . .   | 1349 |
| 7.79.3.51 nppiWarpPerspectiveBack_8u_P4R . . . . .   | 1350 |
| 7.79.3.52 nppiWarpPerspectiveQuad_16u_AC4R . . . . . | 1350 |
| 7.79.3.53 nppiWarpPerspectiveQuad_16u_C1R . . . . .  | 1351 |
| 7.79.3.54 nppiWarpPerspectiveQuad_16u_C3R . . . . .  | 1351 |
| 7.79.3.55 nppiWarpPerspectiveQuad_16u_C4R . . . . .  | 1352 |
| 7.79.3.56 nppiWarpPerspectiveQuad_16u_P3R . . . . .  | 1352 |
| 7.79.3.57 nppiWarpPerspectiveQuad_16u_P4R . . . . .  | 1353 |
| 7.79.3.58 nppiWarpPerspectiveQuad_32f_AC4R . . . . . | 1353 |
| 7.79.3.59 nppiWarpPerspectiveQuad_32f_C1R . . . . .  | 1354 |
| 7.79.3.60 nppiWarpPerspectiveQuad_32f_C3R . . . . .  | 1354 |
| 7.79.3.61 nppiWarpPerspectiveQuad_32f_C4R . . . . .  | 1355 |
| 7.79.3.62 nppiWarpPerspectiveQuad_32f_P3R . . . . .  | 1355 |
| 7.79.3.63 nppiWarpPerspectiveQuad_32f_P4R . . . . .  | 1356 |

|           |  |      |
|-----------|--|------|
| 7.79.3.64 | <a href="#">nppiWarpPerspectiveQuad_32s_AC4R</a> | 1356 |
| 7.79.3.65 | <a href="#">nppiWarpPerspectiveQuad_32s_C1R</a>  | 1357 |
| 7.79.3.66 | <a href="#">nppiWarpPerspectiveQuad_32s_C3R</a>  | 1357 |
| 7.79.3.67 | <a href="#">nppiWarpPerspectiveQuad_32s_C4R</a>  | 1358 |
| 7.79.3.68 | <a href="#">nppiWarpPerspectiveQuad_32s_P3R</a>  | 1358 |
| 7.79.3.69 | <a href="#">nppiWarpPerspectiveQuad_32s_P4R</a>  | 1359 |
| 7.79.3.70 | <a href="#">nppiWarpPerspectiveQuad_8u_AC4R</a>  | 1359 |
| 7.79.3.71 | <a href="#">nppiWarpPerspectiveQuad_8u_C1R</a>   | 1360 |
| 7.79.3.72 | <a href="#">nppiWarpPerspectiveQuad_8u_C3R</a>   | 1360 |
| 7.79.3.73 | <a href="#">nppiWarpPerspectiveQuad_8u_C4R</a>   | 1361 |
| 7.79.3.74 | <a href="#">nppiWarpPerspectiveQuad_8u_P3R</a>   | 1361 |
| 7.79.3.75 | <a href="#">nppiWarpPerspectiveQuad_8u_P4R</a>   | 1362 |
| 7.80      | <a href="#">Linear Transforms</a>                | 1363 |
| 7.80.1    | <a href="#">Detailed Description</a>             | 1363 |
| 7.81      | <a href="#">Fourier Transforms</a>               | 1364 |
| 7.81.1    | <a href="#">Function Documentation</a>           | 1364 |
| 7.81.1.1  | <a href="#">nppiMagnitude_32fc32f_C1R</a>        | 1364 |
| 7.81.1.2  | <a href="#">nppiMagnitudeSqr_32fc32f_C1R</a>     | 1364 |
| 7.82      | <a href="#">Morphological Operations</a>         | 1366 |
| 7.82.1    | <a href="#">Detailed Description</a>             | 1366 |
| 7.83      | <a href="#">Dilation</a>                         | 1367 |
| 7.83.1    | <a href="#">Detailed Description</a>             | 1368 |
| 7.83.2    | <a href="#">Function Documentation</a>           | 1368 |
| 7.83.2.1  | <a href="#">nppiDilate_16u_AC4R</a>              | 1368 |
| 7.83.2.2  | <a href="#">nppiDilate_16u_C1R</a>               | 1368 |
| 7.83.2.3  | <a href="#">nppiDilate_16u_C3R</a>               | 1369 |
| 7.83.2.4  | <a href="#">nppiDilate_16u_C4R</a>               | 1369 |
| 7.83.2.5  | <a href="#">nppiDilate_32f_AC4R</a>              | 1370 |
| 7.83.2.6  | <a href="#">nppiDilate_32f_C1R</a>               | 1370 |
| 7.83.2.7  | <a href="#">nppiDilate_32f_C3R</a>               | 1370 |
| 7.83.2.8  | <a href="#">nppiDilate_32f_C4R</a>               | 1371 |
| 7.83.2.9  | <a href="#">nppiDilate_8u_AC4R</a>               | 1371 |
| 7.83.2.10 | <a href="#">nppiDilate_8u_C1R</a>                | 1372 |
| 7.83.2.11 | <a href="#">nppiDilate_8u_C3R</a>                | 1372 |
| 7.83.2.12 | <a href="#">nppiDilate_8u_C4R</a>                | 1373 |
| 7.84      | <a href="#">Erode</a>                            | 1374 |

|           |                        |      |
|-----------|------------------------|------|
| 7.84.1    | Detailed Description   | 1375 |
| 7.84.2    | Function Documentation | 1375 |
| 7.84.2.1  | npErode_16u_AC4R       | 1375 |
| 7.84.2.2  | npErode_16u_C1R        | 1375 |
| 7.84.2.3  | npErode_16u_C3R        | 1376 |
| 7.84.2.4  | npErode_16u_C4R        | 1376 |
| 7.84.2.5  | npErode_32f_AC4R       | 1377 |
| 7.84.2.6  | npErode_32f_C1R        | 1377 |
| 7.84.2.7  | npErode_32f_C3R        | 1377 |
| 7.84.2.8  | npErode_32f_C4R        | 1378 |
| 7.84.2.9  | npErode_8u_AC4R        | 1378 |
| 7.84.2.10 | npErode_8u_C1R         | 1379 |
| 7.84.2.11 | npErode_8u_C3R         | 1379 |
| 7.84.2.12 | npErode_8u_C4R         | 1380 |
| 7.85      | Dilate3x3              | 1381 |
| 7.85.1    | Detailed Description   | 1382 |
| 7.85.2    | Function Documentation | 1382 |
| 7.85.2.1  | npDilate3x3_16u_AC4R   | 1382 |
| 7.85.2.2  | npDilate3x3_16u_C1R    | 1382 |
| 7.85.2.3  | npDilate3x3_16u_C3R    | 1383 |
| 7.85.2.4  | npDilate3x3_16u_C4R    | 1383 |
| 7.85.2.5  | npDilate3x3_32f_AC4R   | 1383 |
| 7.85.2.6  | npDilate3x3_32f_C1R    | 1384 |
| 7.85.2.7  | npDilate3x3_32f_C3R    | 1384 |
| 7.85.2.8  | npDilate3x3_32f_C4R    | 1384 |
| 7.85.2.9  | npDilate3x3_64f_C1R    | 1385 |
| 7.85.2.10 | npDilate3x3_8u_AC4R    | 1385 |
| 7.85.2.11 | npDilate3x3_8u_C1R     | 1385 |
| 7.85.2.12 | npDilate3x3_8u_C3R     | 1386 |
| 7.85.2.13 | npDilate3x3_8u_C4R     | 1386 |
| 7.86      | Erode3x3               | 1387 |
| 7.86.1    | Detailed Description   | 1388 |
| 7.86.2    | Function Documentation | 1388 |
| 7.86.2.1  | npErode3x3_16u_AC4R    | 1388 |
| 7.86.2.2  | npErode3x3_16u_C1R     | 1388 |
| 7.86.2.3  | npErode3x3_16u_C3R     | 1389 |

|           |  |      |
|-----------|--|------|
| 7.86.2.4  | <a href="#">nppiErode3x3_16u_C4R</a>                       | 1389 |
| 7.86.2.5  | <a href="#">nppiErode3x3_32f_AC4R</a>                      | 1389 |
| 7.86.2.6  | <a href="#">nppiErode3x3_32f_C1R</a>                       | 1390 |
| 7.86.2.7  | <a href="#">nppiErode3x3_32f_C3R</a>                       | 1390 |
| 7.86.2.8  | <a href="#">nppiErode3x3_32f_C4R</a>                       | 1390 |
| 7.86.2.9  | <a href="#">nppiErode3x3_64f_C1R</a>                       | 1391 |
| 7.86.2.10 | <a href="#">nppiErode3x3_8u_AC4R</a>                       | 1391 |
| 7.86.2.11 | <a href="#">nppiErode3x3_8u_C1R</a>                        | 1391 |
| 7.86.2.12 | <a href="#">nppiErode3x3_8u_C3R</a>                        | 1392 |
| 7.86.2.13 | <a href="#">nppiErode3x3_8u_C4R</a>                        | 1392 |
| 7.87      | <a href="#">Statistical Operations</a>                     | 1393 |
| 7.87.1    | <a href="#">Detailed Description</a>                       | 1409 |
| 7.87.2    | <a href="#">Function Documentation</a>                     | 1409 |
| 7.87.2.1  | <a href="#">nppiAverageErrorGetBufferHostSize_16s_C1R</a>  | 1409 |
| 7.87.2.2  | <a href="#">nppiAverageErrorGetBufferHostSize_16s_C2R</a>  | 1409 |
| 7.87.2.3  | <a href="#">nppiAverageErrorGetBufferHostSize_16s_C3R</a>  | 1409 |
| 7.87.2.4  | <a href="#">nppiAverageErrorGetBufferHostSize_16s_C4R</a>  | 1410 |
| 7.87.2.5  | <a href="#">nppiAverageErrorGetBufferHostSize_16sc_C1R</a> | 1410 |
| 7.87.2.6  | <a href="#">nppiAverageErrorGetBufferHostSize_16sc_C2R</a> | 1410 |
| 7.87.2.7  | <a href="#">nppiAverageErrorGetBufferHostSize_16sc_C3R</a> | 1410 |
| 7.87.2.8  | <a href="#">nppiAverageErrorGetBufferHostSize_16sc_C4R</a> | 1411 |
| 7.87.2.9  | <a href="#">nppiAverageErrorGetBufferHostSize_16u_C1R</a>  | 1411 |
| 7.87.2.10 | <a href="#">nppiAverageErrorGetBufferHostSize_16u_C2R</a>  | 1411 |
| 7.87.2.11 | <a href="#">nppiAverageErrorGetBufferHostSize_16u_C3R</a>  | 1412 |
| 7.87.2.12 | <a href="#">nppiAverageErrorGetBufferHostSize_16u_C4R</a>  | 1412 |
| 7.87.2.13 | <a href="#">nppiAverageErrorGetBufferHostSize_32f_C1R</a>  | 1412 |
| 7.87.2.14 | <a href="#">nppiAverageErrorGetBufferHostSize_32f_C2R</a>  | 1412 |
| 7.87.2.15 | <a href="#">nppiAverageErrorGetBufferHostSize_32f_C3R</a>  | 1413 |
| 7.87.2.16 | <a href="#">nppiAverageErrorGetBufferHostSize_32f_C4R</a>  | 1413 |
| 7.87.2.17 | <a href="#">nppiAverageErrorGetBufferHostSize_32fc_C1R</a> | 1413 |
| 7.87.2.18 | <a href="#">nppiAverageErrorGetBufferHostSize_32fc_C2R</a> | 1414 |
| 7.87.2.19 | <a href="#">nppiAverageErrorGetBufferHostSize_32fc_C3R</a> | 1414 |
| 7.87.2.20 | <a href="#">nppiAverageErrorGetBufferHostSize_32fc_C4R</a> | 1414 |
| 7.87.2.21 | <a href="#">nppiAverageErrorGetBufferHostSize_32s_C1R</a>  | 1414 |
| 7.87.2.22 | <a href="#">nppiAverageErrorGetBufferHostSize_32s_C2R</a>  | 1415 |
| 7.87.2.23 | <a href="#">nppiAverageErrorGetBufferHostSize_32s_C3R</a>  | 1415 |



|           |  |      |
|-----------|--|------|
| 7.87.2.24 | <a href="#">nppiAverageErrorGetBufferHostSize_32s_C4R</a>          | 1415 |
| 7.87.2.25 | <a href="#">nppiAverageErrorGetBufferHostSize_32sc_C1R</a>         | 1416 |
| 7.87.2.26 | <a href="#">nppiAverageErrorGetBufferHostSize_32sc_C2R</a>         | 1416 |
| 7.87.2.27 | <a href="#">nppiAverageErrorGetBufferHostSize_32sc_C3R</a>         | 1416 |
| 7.87.2.28 | <a href="#">nppiAverageErrorGetBufferHostSize_32sc_C4R</a>         | 1416 |
| 7.87.2.29 | <a href="#">nppiAverageErrorGetBufferHostSize_32u_C1R</a>          | 1417 |
| 7.87.2.30 | <a href="#">nppiAverageErrorGetBufferHostSize_32u_C2R</a>          | 1417 |
| 7.87.2.31 | <a href="#">nppiAverageErrorGetBufferHostSize_32u_C3R</a>          | 1417 |
| 7.87.2.32 | <a href="#">nppiAverageErrorGetBufferHostSize_32u_C4R</a>          | 1418 |
| 7.87.2.33 | <a href="#">nppiAverageErrorGetBufferHostSize_64f_C1R</a>          | 1418 |
| 7.87.2.34 | <a href="#">nppiAverageErrorGetBufferHostSize_64f_C2R</a>          | 1418 |
| 7.87.2.35 | <a href="#">nppiAverageErrorGetBufferHostSize_64f_C3R</a>          | 1418 |
| 7.87.2.36 | <a href="#">nppiAverageErrorGetBufferHostSize_64f_C4R</a>          | 1419 |
| 7.87.2.37 | <a href="#">nppiAverageErrorGetBufferHostSize_8s_C1R</a>           | 1419 |
| 7.87.2.38 | <a href="#">nppiAverageErrorGetBufferHostSize_8s_C2R</a>           | 1419 |
| 7.87.2.39 | <a href="#">nppiAverageErrorGetBufferHostSize_8s_C3R</a>           | 1420 |
| 7.87.2.40 | <a href="#">nppiAverageErrorGetBufferHostSize_8s_C4R</a>           | 1420 |
| 7.87.2.41 | <a href="#">nppiAverageErrorGetBufferHostSize_8u_C1R</a>           | 1420 |
| 7.87.2.42 | <a href="#">nppiAverageErrorGetBufferHostSize_8u_C2R</a>           | 1420 |
| 7.87.2.43 | <a href="#">nppiAverageErrorGetBufferHostSize_8u_C3R</a>           | 1421 |
| 7.87.2.44 | <a href="#">nppiAverageErrorGetBufferHostSize_8u_C4R</a>           | 1421 |
| 7.87.2.45 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16s_C1R</a>  | 1421 |
| 7.87.2.46 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16s_C2R</a>  | 1422 |
| 7.87.2.47 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16s_C3R</a>  | 1422 |
| 7.87.2.48 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16s_C4R</a>  | 1422 |
| 7.87.2.49 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16sc_C1R</a> | 1422 |
| 7.87.2.50 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16sc_C2R</a> | 1423 |
| 7.87.2.51 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R</a> | 1423 |
| 7.87.2.52 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16sc_C4R</a> | 1423 |
| 7.87.2.53 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16u_C1R</a>  | 1424 |
| 7.87.2.54 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16u_C2R</a>  | 1424 |
| 7.87.2.55 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16u_C3R</a>  | 1424 |
| 7.87.2.56 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_16u_C4R</a>  | 1424 |
| 7.87.2.57 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_32f_C1R</a>  | 1425 |
| 7.87.2.58 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_32f_C2R</a>  | 1425 |
| 7.87.2.59 | <a href="#">nppiAverageRelativeErrorGetBufferHostSize_32f_C3R</a>  | 1425 |

|  |      |
|--|------|
| 7.87.2.60 nppiAverageRelativeErrorGetBufferHostSize_32f_C4R . . . . .  | 1426 |
| 7.87.2.61 nppiAverageRelativeErrorGetBufferHostSize_32fc_C1R . . . . . | 1426 |
| 7.87.2.62 nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R . . . . . | 1426 |
| 7.87.2.63 nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R . . . . . | 1426 |
| 7.87.2.64 nppiAverageRelativeErrorGetBufferHostSize_32fc_C4R . . . . . | 1427 |
| 7.87.2.65 nppiAverageRelativeErrorGetBufferHostSize_32s_C1R . . . . .  | 1427 |
| 7.87.2.66 nppiAverageRelativeErrorGetBufferHostSize_32s_C2R . . . . .  | 1427 |
| 7.87.2.67 nppiAverageRelativeErrorGetBufferHostSize_32s_C3R . . . . .  | 1428 |
| 7.87.2.68 nppiAverageRelativeErrorGetBufferHostSize_32s_C4R . . . . .  | 1428 |
| 7.87.2.69 nppiAverageRelativeErrorGetBufferHostSize_32sc_C1R . . . . . | 1428 |
| 7.87.2.70 nppiAverageRelativeErrorGetBufferHostSize_32sc_C2R . . . . . | 1428 |
| 7.87.2.71 nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R . . . . . | 1429 |
| 7.87.2.72 nppiAverageRelativeErrorGetBufferHostSize_32sc_C4R . . . . . | 1429 |
| 7.87.2.73 nppiAverageRelativeErrorGetBufferHostSize_32u_C1R . . . . .  | 1429 |
| 7.87.2.74 nppiAverageRelativeErrorGetBufferHostSize_32u_C2R . . . . .  | 1430 |
| 7.87.2.75 nppiAverageRelativeErrorGetBufferHostSize_32u_C3R . . . . .  | 1430 |
| 7.87.2.76 nppiAverageRelativeErrorGetBufferHostSize_32u_C4R . . . . .  | 1430 |
| 7.87.2.77 nppiAverageRelativeErrorGetBufferHostSize_64f_C1R . . . . .  | 1430 |
| 7.87.2.78 nppiAverageRelativeErrorGetBufferHostSize_64f_C2R . . . . .  | 1431 |
| 7.87.2.79 nppiAverageRelativeErrorGetBufferHostSize_64f_C3R . . . . .  | 1431 |
| 7.87.2.80 nppiAverageRelativeErrorGetBufferHostSize_64f_C4R . . . . .  | 1431 |
| 7.87.2.81 nppiAverageRelativeErrorGetBufferHostSize_8s_C1R . . . . .   | 1432 |
| 7.87.2.82 nppiAverageRelativeErrorGetBufferHostSize_8s_C2R . . . . .   | 1432 |
| 7.87.2.83 nppiAverageRelativeErrorGetBufferHostSize_8s_C3R . . . . .   | 1432 |
| 7.87.2.84 nppiAverageRelativeErrorGetBufferHostSize_8s_C4R . . . . .   | 1432 |
| 7.87.2.85 nppiAverageRelativeErrorGetBufferHostSize_8u_C1R . . . . .   | 1433 |
| 7.87.2.86 nppiAverageRelativeErrorGetBufferHostSize_8u_C2R . . . . .   | 1433 |
| 7.87.2.87 nppiAverageRelativeErrorGetBufferHostSize_8u_C3R . . . . .   | 1433 |
| 7.87.2.88 nppiAverageRelativeErrorGetBufferHostSize_8u_C4R . . . . .   | 1434 |
| 7.87.2.89 nppiMaximumErrorGetBufferHostSize_16s_C1R . . . . .          | 1434 |
| 7.87.2.90 nppiMaximumErrorGetBufferHostSize_16s_C2R . . . . .          | 1434 |
| 7.87.2.91 nppiMaximumErrorGetBufferHostSize_16s_C3R . . . . .          | 1434 |
| 7.87.2.92 nppiMaximumErrorGetBufferHostSize_16s_C4R . . . . .          | 1435 |
| 7.87.2.93 nppiMaximumErrorGetBufferHostSize_16sc_C1R . . . . .         | 1435 |
| 7.87.2.94 nppiMaximumErrorGetBufferHostSize_16sc_C2R . . . . .         | 1435 |
| 7.87.2.95 nppiMaximumErrorGetBufferHostSize_16sc_C3R . . . . .         | 1436 |

|            |  |      |
|------------|--|------|
| 7.87.2.96  | <a href="#">nppiMaximumErrorGetBufferHostSize_16sc_C4R</a> | 1436 |
| 7.87.2.97  | <a href="#">nppiMaximumErrorGetBufferHostSize_16u_C1R</a>  | 1436 |
| 7.87.2.98  | <a href="#">nppiMaximumErrorGetBufferHostSize_16u_C2R</a>  | 1436 |
| 7.87.2.99  | <a href="#">nppiMaximumErrorGetBufferHostSize_16u_C3R</a>  | 1437 |
| 7.87.2.100 | <a href="#">nppiMaximumErrorGetBufferHostSize_16u_C4R</a>  | 1437 |
| 7.87.2.101 | <a href="#">nppiMaximumErrorGetBufferHostSize_32f_C1R</a>  | 1437 |
| 7.87.2.102 | <a href="#">nppiMaximumErrorGetBufferHostSize_32f_C2R</a>  | 1438 |
| 7.87.2.103 | <a href="#">nppiMaximumErrorGetBufferHostSize_32f_C3R</a>  | 1438 |
| 7.87.2.104 | <a href="#">nppiMaximumErrorGetBufferHostSize_32f_C4R</a>  | 1438 |
| 7.87.2.105 | <a href="#">nppiMaximumErrorGetBufferHostSize_32fc_C1R</a> | 1438 |
| 7.87.2.106 | <a href="#">nppiMaximumErrorGetBufferHostSize_32fc_C2R</a> | 1439 |
| 7.87.2.107 | <a href="#">nppiMaximumErrorGetBufferHostSize_32fc_C3R</a> | 1439 |
| 7.87.2.108 | <a href="#">nppiMaximumErrorGetBufferHostSize_32fc_C4R</a> | 1439 |
| 7.87.2.109 | <a href="#">nppiMaximumErrorGetBufferHostSize_32s_C1R</a>  | 1440 |
| 7.87.2.110 | <a href="#">nppiMaximumErrorGetBufferHostSize_32s_C2R</a>  | 1440 |
| 7.87.2.111 | <a href="#">nppiMaximumErrorGetBufferHostSize_32s_C3R</a>  | 1440 |
| 7.87.2.112 | <a href="#">nppiMaximumErrorGetBufferHostSize_32s_C4R</a>  | 1440 |
| 7.87.2.113 | <a href="#">nppiMaximumErrorGetBufferHostSize_32sc_C1R</a> | 1441 |
| 7.87.2.114 | <a href="#">nppiMaximumErrorGetBufferHostSize_32sc_C2R</a> | 1441 |
| 7.87.2.115 | <a href="#">nppiMaximumErrorGetBufferHostSize_32sc_C3R</a> | 1441 |
| 7.87.2.116 | <a href="#">nppiMaximumErrorGetBufferHostSize_32sc_C4R</a> | 1442 |
| 7.87.2.117 | <a href="#">nppiMaximumErrorGetBufferHostSize_32u_C1R</a>  | 1442 |
| 7.87.2.118 | <a href="#">nppiMaximumErrorGetBufferHostSize_32u_C2R</a>  | 1442 |
| 7.87.2.119 | <a href="#">nppiMaximumErrorGetBufferHostSize_32u_C3R</a>  | 1442 |
| 7.87.2.120 | <a href="#">nppiMaximumErrorGetBufferHostSize_32u_C4R</a>  | 1443 |
| 7.87.2.121 | <a href="#">nppiMaximumErrorGetBufferHostSize_64f_C1R</a>  | 1443 |
| 7.87.2.122 | <a href="#">nppiMaximumErrorGetBufferHostSize_64f_C2R</a>  | 1443 |
| 7.87.2.123 | <a href="#">nppiMaximumErrorGetBufferHostSize_64f_C3R</a>  | 1444 |
| 7.87.2.124 | <a href="#">nppiMaximumErrorGetBufferHostSize_64f_C4R</a>  | 1444 |
| 7.87.2.125 | <a href="#">nppiMaximumErrorGetBufferHostSize_8s_C1R</a>   | 1444 |
| 7.87.2.126 | <a href="#">nppiMaximumErrorGetBufferHostSize_8s_C2R</a>   | 1444 |
| 7.87.2.127 | <a href="#">nppiMaximumErrorGetBufferHostSize_8s_C3R</a>   | 1445 |
| 7.87.2.128 | <a href="#">nppiMaximumErrorGetBufferHostSize_8s_C4R</a>   | 1445 |
| 7.87.2.129 | <a href="#">nppiMaximumErrorGetBufferHostSize_8u_C1R</a>   | 1445 |
| 7.87.2.130 | <a href="#">nppiMaximumErrorGetBufferHostSize_8u_C2R</a>   | 1446 |
| 7.87.2.131 | <a href="#">nppiMaximumErrorGetBufferHostSize_8u_C3R</a>   | 1446 |

|  |      |
|--|------|
| 7.87.2.132nppiMaximumErrorGetBufferHostSize_8u_C4R . . . . .           | 1446 |
| 7.87.2.133nppiMaximumRelativeErrorGetBufferHostSize_16s_C1R . . . . .  | 1446 |
| 7.87.2.134nppiMaximumRelativeErrorGetBufferHostSize_16s_C2R . . . . .  | 1447 |
| 7.87.2.135nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R . . . . .  | 1447 |
| 7.87.2.136nppiMaximumRelativeErrorGetBufferHostSize_16s_C4R . . . . .  | 1447 |
| 7.87.2.137nppiMaximumRelativeErrorGetBufferHostSize_16sc_C1R . . . . . | 1448 |
| 7.87.2.138nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R . . . . . | 1448 |
| 7.87.2.139nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R . . . . . | 1448 |
| 7.87.2.140nppiMaximumRelativeErrorGetBufferHostSize_16sc_C4R . . . . . | 1448 |
| 7.87.2.141nppiMaximumRelativeErrorGetBufferHostSize_16u_C1R . . . . .  | 1449 |
| 7.87.2.142nppiMaximumRelativeErrorGetBufferHostSize_16u_C2R . . . . .  | 1449 |
| 7.87.2.143nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R . . . . .  | 1449 |
| 7.87.2.144nppiMaximumRelativeErrorGetBufferHostSize_16u_C4R . . . . .  | 1450 |
| 7.87.2.145nppiMaximumRelativeErrorGetBufferHostSize_32f_C1R . . . . .  | 1450 |
| 7.87.2.146nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R . . . . .  | 1450 |
| 7.87.2.147nppiMaximumRelativeErrorGetBufferHostSize_32f_C3R . . . . .  | 1450 |
| 7.87.2.148nppiMaximumRelativeErrorGetBufferHostSize_32f_C4R . . . . .  | 1451 |
| 7.87.2.149nppiMaximumRelativeErrorGetBufferHostSize_32fc_C1R . . . . . | 1451 |
| 7.87.2.150nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R . . . . . | 1451 |
| 7.87.2.151nppiMaximumRelativeErrorGetBufferHostSize_32fc_C3R . . . . . | 1452 |
| 7.87.2.152nppiMaximumRelativeErrorGetBufferHostSize_32fc_C4R . . . . . | 1452 |
| 7.87.2.153nppiMaximumRelativeErrorGetBufferHostSize_32s_C1R . . . . .  | 1452 |
| 7.87.2.154nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R . . . . .  | 1452 |
| 7.87.2.155nppiMaximumRelativeErrorGetBufferHostSize_32s_C3R . . . . .  | 1453 |
| 7.87.2.156nppiMaximumRelativeErrorGetBufferHostSize_32s_C4R . . . . .  | 1453 |
| 7.87.2.157nppiMaximumRelativeErrorGetBufferHostSize_32sc_C1R . . . . . | 1453 |
| 7.87.2.158nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R . . . . . | 1454 |
| 7.87.2.159nppiMaximumRelativeErrorGetBufferHostSize_32sc_C3R . . . . . | 1454 |
| 7.87.2.160nppiMaximumRelativeErrorGetBufferHostSize_32sc_C4R . . . . . | 1454 |
| 7.87.2.161nppiMaximumRelativeErrorGetBufferHostSize_32u_C1R . . . . .  | 1454 |
| 7.87.2.162nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R . . . . .  | 1455 |
| 7.87.2.163nppiMaximumRelativeErrorGetBufferHostSize_32u_C3R . . . . .  | 1455 |
| 7.87.2.164nppiMaximumRelativeErrorGetBufferHostSize_32u_C4R . . . . .  | 1455 |
| 7.87.2.165nppiMaximumRelativeErrorGetBufferHostSize_64f_C1R . . . . .  | 1456 |
| 7.87.2.166nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R . . . . .  | 1456 |
| 7.87.2.167nppiMaximumRelativeErrorGetBufferHostSize_64f_C3R . . . . .  | 1456 |

|            |   |      |
|------------|---|------|
| 7.87.2.168 | nppiMaximumRelativeErrorGetBufferHostSize_64f_C4R | 1456 |
| 7.87.2.169 | nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R  | 1457 |
| 7.87.2.170 | nppiMaximumRelativeErrorGetBufferHostSize_8s_C2R  | 1457 |
| 7.87.2.171 | nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R  | 1457 |
| 7.87.2.172 | nppiMaximumRelativeErrorGetBufferHostSize_8s_C4R  | 1458 |
| 7.87.2.173 | nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R  | 1458 |
| 7.87.2.174 | nppiMaximumRelativeErrorGetBufferHostSize_8u_C2R  | 1458 |
| 7.87.2.175 | nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R  | 1458 |
| 7.87.2.176 | nppiMaximumRelativeErrorGetBufferHostSize_8u_C4R  | 1459 |
| 7.88       | Sum   | 1460 |
| 7.88.1     | Detailed Description                              | 1462 |
| 7.88.2     | Function Documentation                            | 1463 |
| 7.88.2.1   | nppiSum_16s_AC4R                                  | 1463 |
| 7.88.2.2   | nppiSum_16s_C1R                                   | 1463 |
| 7.88.2.3   | nppiSum_16s_C3R                                   | 1463 |
| 7.88.2.4   | nppiSum_16s_C4R                                   | 1464 |
| 7.88.2.5   | nppiSum_16u_AC4R                                  | 1464 |
| 7.88.2.6   | nppiSum_16u_C1R                                   | 1464 |
| 7.88.2.7   | nppiSum_16u_C3R                                   | 1465 |
| 7.88.2.8   | nppiSum_16u_C4R                                   | 1465 |
| 7.88.2.9   | nppiSum_32f_AC4R                                  | 1466 |
| 7.88.2.10  | nppiSum_32f_C1R                                   | 1466 |
| 7.88.2.11  | nppiSum_32f_C3R                                   | 1466 |
| 7.88.2.12  | nppiSum_32f_C4R                                   | 1467 |
| 7.88.2.13  | nppiSum_8u64s_C1R                                 | 1467 |
| 7.88.2.14  | nppiSum_8u64s_C4R                                 | 1467 |
| 7.88.2.15  | nppiSum_8u_AC4R                                   | 1468 |
| 7.88.2.16  | nppiSum_8u_C1R                                    | 1468 |
| 7.88.2.17  | nppiSum_8u_C3R                                    | 1469 |
| 7.88.2.18  | nppiSum_8u_C4R                                    | 1469 |
| 7.88.2.19  | nppiSumGetBufferHostSize_16s_AC4R                 | 1469 |
| 7.88.2.20  | nppiSumGetBufferHostSize_16s_C1R                  | 1470 |
| 7.88.2.21  | nppiSumGetBufferHostSize_16s_C3R                  | 1470 |
| 7.88.2.22  | nppiSumGetBufferHostSize_16s_C4R                  | 1470 |
| 7.88.2.23  | nppiSumGetBufferHostSize_16u_AC4R                 | 1470 |
| 7.88.2.24  | nppiSumGetBufferHostSize_16u_C1R                  | 1471 |

|           |  |      |
|-----------|--|------|
| 7.88.2.25 | <a href="#">nppiSumGetBufferHostSize_16u_C3R</a>   | 1471 |
| 7.88.2.26 | <a href="#">nppiSumGetBufferHostSize_16u_C4R</a>   | 1471 |
| 7.88.2.27 | <a href="#">nppiSumGetBufferHostSize_32f_AC4R</a>  | 1472 |
| 7.88.2.28 | <a href="#">nppiSumGetBufferHostSize_32f_C1R</a>   | 1472 |
| 7.88.2.29 | <a href="#">nppiSumGetBufferHostSize_32f_C3R</a>   | 1472 |
| 7.88.2.30 | <a href="#">nppiSumGetBufferHostSize_32f_C4R</a>   | 1472 |
| 7.88.2.31 | <a href="#">nppiSumGetBufferHostSize_8u64s_C1R</a> | 1473 |
| 7.88.2.32 | <a href="#">nppiSumGetBufferHostSize_8u64s_C4R</a> | 1473 |
| 7.88.2.33 | <a href="#">nppiSumGetBufferHostSize_8u_AC4R</a>   | 1473 |
| 7.88.2.34 | <a href="#">nppiSumGetBufferHostSize_8u_C1R</a>    | 1474 |
| 7.88.2.35 | <a href="#">nppiSumGetBufferHostSize_8u_C3R</a>    | 1474 |
| 7.88.2.36 | <a href="#">nppiSumGetBufferHostSize_8u_C4R</a>    | 1474 |
| 7.89      | <a href="#">Min</a>                                | 1475 |
| 7.89.1    | <a href="#">Detailed Description</a>               | 1477 |
| 7.89.2    | <a href="#">Function Documentation</a>             | 1477 |
| 7.89.2.1  | <a href="#">nppiMin_16s_AC4R</a>                   | 1477 |
| 7.89.2.2  | <a href="#">nppiMin_16s_C1R</a>                    | 1478 |
| 7.89.2.3  | <a href="#">nppiMin_16s_C3R</a>                    | 1478 |
| 7.89.2.4  | <a href="#">nppiMin_16s_C4R</a>                    | 1478 |
| 7.89.2.5  | <a href="#">nppiMin_16u_AC4R</a>                   | 1479 |
| 7.89.2.6  | <a href="#">nppiMin_16u_C1R</a>                    | 1479 |
| 7.89.2.7  | <a href="#">nppiMin_16u_C3R</a>                    | 1479 |
| 7.89.2.8  | <a href="#">nppiMin_16u_C4R</a>                    | 1480 |
| 7.89.2.9  | <a href="#">nppiMin_32f_AC4R</a>                   | 1480 |
| 7.89.2.10 | <a href="#">nppiMin_32f_C1R</a>                    | 1480 |
| 7.89.2.11 | <a href="#">nppiMin_32f_C3R</a>                    | 1481 |
| 7.89.2.12 | <a href="#">nppiMin_32f_C4R</a>                    | 1481 |
| 7.89.2.13 | <a href="#">nppiMin_8u_AC4R</a>                    | 1482 |
| 7.89.2.14 | <a href="#">nppiMin_8u_C1R</a>                     | 1482 |
| 7.89.2.15 | <a href="#">nppiMin_8u_C3R</a>                     | 1482 |
| 7.89.2.16 | <a href="#">nppiMin_8u_C4R</a>                     | 1483 |
| 7.89.2.17 | <a href="#">nppiMinGetBufferHostSize_16s_AC4R</a>  | 1483 |
| 7.89.2.18 | <a href="#">nppiMinGetBufferHostSize_16s_C1R</a>   | 1483 |
| 7.89.2.19 | <a href="#">nppiMinGetBufferHostSize_16s_C3R</a>   | 1484 |
| 7.89.2.20 | <a href="#">nppiMinGetBufferHostSize_16s_C4R</a>   | 1484 |
| 7.89.2.21 | <a href="#">nppiMinGetBufferHostSize_16u_AC4R</a>  | 1484 |

|           |   |      |
|-----------|---|------|
| 7.89.2.22 | <a href="#">nppiMinGetBufferHostSize_16u_C1R</a>      | 1484 |
| 7.89.2.23 | <a href="#">nppiMinGetBufferHostSize_16u_C3R</a>      | 1485 |
| 7.89.2.24 | <a href="#">nppiMinGetBufferHostSize_16u_C4R</a>      | 1485 |
| 7.89.2.25 | <a href="#">nppiMinGetBufferHostSize_32f_AC4R</a>     | 1485 |
| 7.89.2.26 | <a href="#">nppiMinGetBufferHostSize_32f_C1R</a>      | 1486 |
| 7.89.2.27 | <a href="#">nppiMinGetBufferHostSize_32f_C3R</a>      | 1486 |
| 7.89.2.28 | <a href="#">nppiMinGetBufferHostSize_32f_C4R</a>      | 1486 |
| 7.89.2.29 | <a href="#">nppiMinGetBufferHostSize_8u_AC4R</a>      | 1486 |
| 7.89.2.30 | <a href="#">nppiMinGetBufferHostSize_8u_C1R</a>       | 1487 |
| 7.89.2.31 | <a href="#">nppiMinGetBufferHostSize_8u_C3R</a>       | 1487 |
| 7.89.2.32 | <a href="#">nppiMinGetBufferHostSize_8u_C4R</a>       | 1487 |
| 7.90      | <a href="#">MinIndx</a>                               | 1488 |
| 7.90.1    | <a href="#">Detailed Description</a>                  | 1490 |
| 7.90.2    | <a href="#">Function Documentation</a>                | 1490 |
| 7.90.2.1  | <a href="#">nppiMinIndx_16s_AC4R</a>                  | 1490 |
| 7.90.2.2  | <a href="#">nppiMinIndx_16s_C1R</a>                   | 1491 |
| 7.90.2.3  | <a href="#">nppiMinIndx_16s_C3R</a>                   | 1491 |
| 7.90.2.4  | <a href="#">nppiMinIndx_16s_C4R</a>                   | 1492 |
| 7.90.2.5  | <a href="#">nppiMinIndx_16u_AC4R</a>                  | 1492 |
| 7.90.2.6  | <a href="#">nppiMinIndx_16u_C1R</a>                   | 1492 |
| 7.90.2.7  | <a href="#">nppiMinIndx_16u_C3R</a>                   | 1493 |
| 7.90.2.8  | <a href="#">nppiMinIndx_16u_C4R</a>                   | 1493 |
| 7.90.2.9  | <a href="#">nppiMinIndx_32f_AC4R</a>                  | 1494 |
| 7.90.2.10 | <a href="#">nppiMinIndx_32f_C1R</a>                   | 1494 |
| 7.90.2.11 | <a href="#">nppiMinIndx_32f_C3R</a>                   | 1494 |
| 7.90.2.12 | <a href="#">nppiMinIndx_32f_C4R</a>                   | 1495 |
| 7.90.2.13 | <a href="#">nppiMinIndx_8u_AC4R</a>                   | 1495 |
| 7.90.2.14 | <a href="#">nppiMinIndx_8u_C1R</a>                    | 1496 |
| 7.90.2.15 | <a href="#">nppiMinIndx_8u_C3R</a>                    | 1496 |
| 7.90.2.16 | <a href="#">nppiMinIndx_8u_C4R</a>                    | 1496 |
| 7.90.2.17 | <a href="#">nppiMinIndxGetBufferHostSize_16s_AC4R</a> | 1497 |
| 7.90.2.18 | <a href="#">nppiMinIndxGetBufferHostSize_16s_C1R</a>  | 1497 |
| 7.90.2.19 | <a href="#">nppiMinIndxGetBufferHostSize_16s_C3R</a>  | 1497 |
| 7.90.2.20 | <a href="#">nppiMinIndxGetBufferHostSize_16s_C4R</a>  | 1498 |
| 7.90.2.21 | <a href="#">nppiMinIndxGetBufferHostSize_16u_AC4R</a> | 1498 |
| 7.90.2.22 | <a href="#">nppiMinIndxGetBufferHostSize_16u_C1R</a>  | 1498 |

|           |   |      |
|-----------|---|------|
| 7.90.2.23 | <a href="#">nppiMinIndxGetBufferHostSize_16u_C3R</a>  | 1499 |
| 7.90.2.24 | <a href="#">nppiMinIndxGetBufferHostSize_16u_C4R</a>  | 1499 |
| 7.90.2.25 | <a href="#">nppiMinIndxGetBufferHostSize_32f_AC4R</a> | 1499 |
| 7.90.2.26 | <a href="#">nppiMinIndxGetBufferHostSize_32f_C1R</a>  | 1499 |
| 7.90.2.27 | <a href="#">nppiMinIndxGetBufferHostSize_32f_C3R</a>  | 1500 |
| 7.90.2.28 | <a href="#">nppiMinIndxGetBufferHostSize_32f_C4R</a>  | 1500 |
| 7.90.2.29 | <a href="#">nppiMinIndxGetBufferHostSize_8u_AC4R</a>  | 1500 |
| 7.90.2.30 | <a href="#">nppiMinIndxGetBufferHostSize_8u_C1R</a>   | 1501 |
| 7.90.2.31 | <a href="#">nppiMinIndxGetBufferHostSize_8u_C3R</a>   | 1501 |
| 7.90.2.32 | <a href="#">nppiMinIndxGetBufferHostSize_8u_C4R</a>   | 1501 |
| 7.91      | <a href="#">Max</a>                                   | 1502 |
| 7.91.1    | <a href="#">Detailed Description</a>                  | 1504 |
| 7.91.2    | <a href="#">Function Documentation</a>                | 1504 |
| 7.91.2.1  | <a href="#">nppiMax_16s_AC4R</a>                      | 1504 |
| 7.91.2.2  | <a href="#">nppiMax_16s_C1R</a>                       | 1505 |
| 7.91.2.3  | <a href="#">nppiMax_16s_C3R</a>                       | 1505 |
| 7.91.2.4  | <a href="#">nppiMax_16s_C4R</a>                       | 1505 |
| 7.91.2.5  | <a href="#">nppiMax_16u_AC4R</a>                      | 1506 |
| 7.91.2.6  | <a href="#">nppiMax_16u_C1R</a>                       | 1506 |
| 7.91.2.7  | <a href="#">nppiMax_16u_C3R</a>                       | 1506 |
| 7.91.2.8  | <a href="#">nppiMax_16u_C4R</a>                       | 1507 |
| 7.91.2.9  | <a href="#">nppiMax_32f_AC4R</a>                      | 1507 |
| 7.91.2.10 | <a href="#">nppiMax_32f_C1R</a>                       | 1507 |
| 7.91.2.11 | <a href="#">nppiMax_32f_C3R</a>                       | 1508 |
| 7.91.2.12 | <a href="#">nppiMax_32f_C4R</a>                       | 1508 |
| 7.91.2.13 | <a href="#">nppiMax_8u_AC4R</a>                       | 1509 |
| 7.91.2.14 | <a href="#">nppiMax_8u_C1R</a>                        | 1509 |
| 7.91.2.15 | <a href="#">nppiMax_8u_C3R</a>                        | 1509 |
| 7.91.2.16 | <a href="#">nppiMax_8u_C4R</a>                        | 1510 |
| 7.91.2.17 | <a href="#">nppiMaxGetBufferHostSize_16s_AC4R</a>     | 1510 |
| 7.91.2.18 | <a href="#">nppiMaxGetBufferHostSize_16s_C1R</a>      | 1510 |
| 7.91.2.19 | <a href="#">nppiMaxGetBufferHostSize_16s_C3R</a>      | 1511 |
| 7.91.2.20 | <a href="#">nppiMaxGetBufferHostSize_16s_C4R</a>      | 1511 |
| 7.91.2.21 | <a href="#">nppiMaxGetBufferHostSize_16u_AC4R</a>     | 1511 |
| 7.91.2.22 | <a href="#">nppiMaxGetBufferHostSize_16u_C1R</a>      | 1511 |
| 7.91.2.23 | <a href="#">nppiMaxGetBufferHostSize_16u_C3R</a>      | 1512 |



|           |   |      |
|-----------|---|------|
| 7.91.2.24 | <a href="#">nppiMaxGetBufferHostSize_16u_C4R</a>      | 1512 |
| 7.91.2.25 | <a href="#">nppiMaxGetBufferHostSize_32f_AC4R</a>     | 1512 |
| 7.91.2.26 | <a href="#">nppiMaxGetBufferHostSize_32f_C1R</a>      | 1513 |
| 7.91.2.27 | <a href="#">nppiMaxGetBufferHostSize_32f_C3R</a>      | 1513 |
| 7.91.2.28 | <a href="#">nppiMaxGetBufferHostSize_32f_C4R</a>      | 1513 |
| 7.91.2.29 | <a href="#">nppiMaxGetBufferHostSize_8u_AC4R</a>      | 1513 |
| 7.91.2.30 | <a href="#">nppiMaxGetBufferHostSize_8u_C1R</a>       | 1514 |
| 7.91.2.31 | <a href="#">nppiMaxGetBufferHostSize_8u_C3R</a>       | 1514 |
| 7.91.2.32 | <a href="#">nppiMaxGetBufferHostSize_8u_C4R</a>       | 1514 |
| 7.92      | <a href="#">MaxIndx</a>                               | 1515 |
| 7.92.1    | <a href="#">Detailed Description</a>                  | 1517 |
| 7.92.2    | <a href="#">Function Documentation</a>                | 1517 |
| 7.92.2.1  | <a href="#">nppiMaxIndx_16s_AC4R</a>                  | 1517 |
| 7.92.2.2  | <a href="#">nppiMaxIndx_16s_C1R</a>                   | 1518 |
| 7.92.2.3  | <a href="#">nppiMaxIndx_16s_C3R</a>                   | 1518 |
| 7.92.2.4  | <a href="#">nppiMaxIndx_16s_C4R</a>                   | 1519 |
| 7.92.2.5  | <a href="#">nppiMaxIndx_16u_AC4R</a>                  | 1519 |
| 7.92.2.6  | <a href="#">nppiMaxIndx_16u_C1R</a>                   | 1519 |
| 7.92.2.7  | <a href="#">nppiMaxIndx_16u_C3R</a>                   | 1520 |
| 7.92.2.8  | <a href="#">nppiMaxIndx_16u_C4R</a>                   | 1520 |
| 7.92.2.9  | <a href="#">nppiMaxIndx_32f_AC4R</a>                  | 1521 |
| 7.92.2.10 | <a href="#">nppiMaxIndx_32f_C1R</a>                   | 1521 |
| 7.92.2.11 | <a href="#">nppiMaxIndx_32f_C3R</a>                   | 1521 |
| 7.92.2.12 | <a href="#">nppiMaxIndx_32f_C4R</a>                   | 1522 |
| 7.92.2.13 | <a href="#">nppiMaxIndx_8u_AC4R</a>                   | 1522 |
| 7.92.2.14 | <a href="#">nppiMaxIndx_8u_C1R</a>                    | 1523 |
| 7.92.2.15 | <a href="#">nppiMaxIndx_8u_C3R</a>                    | 1523 |
| 7.92.2.16 | <a href="#">nppiMaxIndx_8u_C4R</a>                    | 1523 |
| 7.92.2.17 | <a href="#">nppiMaxIndxGetBufferHostSize_16s_AC4R</a> | 1524 |
| 7.92.2.18 | <a href="#">nppiMaxIndxGetBufferHostSize_16s_C1R</a>  | 1524 |
| 7.92.2.19 | <a href="#">nppiMaxIndxGetBufferHostSize_16s_C3R</a>  | 1524 |
| 7.92.2.20 | <a href="#">nppiMaxIndxGetBufferHostSize_16s_C4R</a>  | 1525 |
| 7.92.2.21 | <a href="#">nppiMaxIndxGetBufferHostSize_16u_AC4R</a> | 1525 |
| 7.92.2.22 | <a href="#">nppiMaxIndxGetBufferHostSize_16u_C1R</a>  | 1525 |
| 7.92.2.23 | <a href="#">nppiMaxIndxGetBufferHostSize_16u_C3R</a>  | 1526 |
| 7.92.2.24 | <a href="#">nppiMaxIndxGetBufferHostSize_16u_C4R</a>  | 1526 |

|           |   |      |
|-----------|---|------|
| 7.92.2.25 | <a href="#">nppiMaxIndxGetBufferHostSize_32f_AC4R</a> | 1526 |
| 7.92.2.26 | <a href="#">nppiMaxIndxGetBufferHostSize_32f_C1R</a>  | 1526 |
| 7.92.2.27 | <a href="#">nppiMaxIndxGetBufferHostSize_32f_C3R</a>  | 1527 |
| 7.92.2.28 | <a href="#">nppiMaxIndxGetBufferHostSize_32f_C4R</a>  | 1527 |
| 7.92.2.29 | <a href="#">nppiMaxIndxGetBufferHostSize_8u_AC4R</a>  | 1527 |
| 7.92.2.30 | <a href="#">nppiMaxIndxGetBufferHostSize_8u_C1R</a>   | 1528 |
| 7.92.2.31 | <a href="#">nppiMaxIndxGetBufferHostSize_8u_C3R</a>   | 1528 |
| 7.92.2.32 | <a href="#">nppiMaxIndxGetBufferHostSize_8u_C4R</a>   | 1528 |
| 7.93      | <a href="#">MinMax</a>                                | 1529 |
| 7.93.1    | <a href="#">Detailed Description</a>                  | 1531 |
| 7.93.2    | <a href="#">Function Documentation</a>                | 1531 |
| 7.93.2.1  | <a href="#">nppiMinMax_16s_AC4R</a>                   | 1531 |
| 7.93.2.2  | <a href="#">nppiMinMax_16s_C1R</a>                    | 1532 |
| 7.93.2.3  | <a href="#">nppiMinMax_16s_C3R</a>                    | 1532 |
| 7.93.2.4  | <a href="#">nppiMinMax_16s_C4R</a>                    | 1532 |
| 7.93.2.5  | <a href="#">nppiMinMax_16u_AC4R</a>                   | 1533 |
| 7.93.2.6  | <a href="#">nppiMinMax_16u_C1R</a>                    | 1533 |
| 7.93.2.7  | <a href="#">nppiMinMax_16u_C3R</a>                    | 1534 |
| 7.93.2.8  | <a href="#">nppiMinMax_16u_C4R</a>                    | 1534 |
| 7.93.2.9  | <a href="#">nppiMinMax_32f_AC4R</a>                   | 1534 |
| 7.93.2.10 | <a href="#">nppiMinMax_32f_C1R</a>                    | 1535 |
| 7.93.2.11 | <a href="#">nppiMinMax_32f_C3R</a>                    | 1535 |
| 7.93.2.12 | <a href="#">nppiMinMax_32f_C4R</a>                    | 1536 |
| 7.93.2.13 | <a href="#">nppiMinMax_8u_AC4R</a>                    | 1536 |
| 7.93.2.14 | <a href="#">nppiMinMax_8u_C1R</a>                     | 1536 |
| 7.93.2.15 | <a href="#">nppiMinMax_8u_C3R</a>                     | 1537 |
| 7.93.2.16 | <a href="#">nppiMinMax_8u_C4R</a>                     | 1537 |
| 7.93.2.17 | <a href="#">nppiMinMaxGetBufferHostSize_16s_AC4R</a>  | 1538 |
| 7.93.2.18 | <a href="#">nppiMinMaxGetBufferHostSize_16s_C1R</a>   | 1538 |
| 7.93.2.19 | <a href="#">nppiMinMaxGetBufferHostSize_16s_C3R</a>   | 1538 |
| 7.93.2.20 | <a href="#">nppiMinMaxGetBufferHostSize_16s_C4R</a>   | 1538 |
| 7.93.2.21 | <a href="#">nppiMinMaxGetBufferHostSize_16u_AC4R</a>  | 1539 |
| 7.93.2.22 | <a href="#">nppiMinMaxGetBufferHostSize_16u_C1R</a>   | 1539 |
| 7.93.2.23 | <a href="#">nppiMinMaxGetBufferHostSize_16u_C3R</a>   | 1539 |
| 7.93.2.24 | <a href="#">nppiMinMaxGetBufferHostSize_16u_C4R</a>   | 1540 |
| 7.93.2.25 | <a href="#">nppiMinMaxGetBufferHostSize_32f_AC4R</a>  | 1540 |

|           |   |      |
|-----------|---|------|
| 7.93.2.26 | <a href="#">nppiMinMaxGetBufferHostSize_32f_C1R</a>       | 1540 |
| 7.93.2.27 | <a href="#">nppiMinMaxGetBufferHostSize_32f_C3R</a>       | 1540 |
| 7.93.2.28 | <a href="#">nppiMinMaxGetBufferHostSize_32f_C4R</a>       | 1541 |
| 7.93.2.29 | <a href="#">nppiMinMaxGetBufferHostSize_8u_AC4R</a>       | 1541 |
| 7.93.2.30 | <a href="#">nppiMinMaxGetBufferHostSize_8u_C1R</a>        | 1541 |
| 7.93.2.31 | <a href="#">nppiMinMaxGetBufferHostSize_8u_C3R</a>        | 1542 |
| 7.93.2.32 | <a href="#">nppiMinMaxGetBufferHostSize_8u_C4R</a>        | 1542 |
| 7.94      | <a href="#">MinMaxIndx</a>                                | 1543 |
| 7.94.1    | <a href="#">Detailed Description</a>                      | 1546 |
| 7.94.2    | <a href="#">Function Documentation</a>                    | 1546 |
| 7.94.2.1  | <a href="#">nppiMinMaxIndx_16u_C1MR</a>                   | 1546 |
| 7.94.2.2  | <a href="#">nppiMinMaxIndx_16u_C1R</a>                    | 1547 |
| 7.94.2.3  | <a href="#">nppiMinMaxIndx_16u_C3CMR</a>                  | 1547 |
| 7.94.2.4  | <a href="#">nppiMinMaxIndx_16u_C3CR</a>                   | 1548 |
| 7.94.2.5  | <a href="#">nppiMinMaxIndx_32f_C1MR</a>                   | 1549 |
| 7.94.2.6  | <a href="#">nppiMinMaxIndx_32f_C1R</a>                    | 1549 |
| 7.94.2.7  | <a href="#">nppiMinMaxIndx_32f_C3CMR</a>                  | 1550 |
| 7.94.2.8  | <a href="#">nppiMinMaxIndx_32f_C3CR</a>                   | 1550 |
| 7.94.2.9  | <a href="#">nppiMinMaxIndx_8s_C1MR</a>                    | 1551 |
| 7.94.2.10 | <a href="#">nppiMinMaxIndx_8s_C1R</a>                     | 1551 |
| 7.94.2.11 | <a href="#">nppiMinMaxIndx_8s_C3CMR</a>                   | 1552 |
| 7.94.2.12 | <a href="#">nppiMinMaxIndx_8s_C3CR</a>                    | 1552 |
| 7.94.2.13 | <a href="#">nppiMinMaxIndx_8u_C1MR</a>                    | 1553 |
| 7.94.2.14 | <a href="#">nppiMinMaxIndx_8u_C1R</a>                     | 1554 |
| 7.94.2.15 | <a href="#">nppiMinMaxIndx_8u_C3CMR</a>                   | 1554 |
| 7.94.2.16 | <a href="#">nppiMinMaxIndx_8u_C3CR</a>                    | 1555 |
| 7.94.2.17 | <a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C1MR</a>  | 1555 |
| 7.94.2.18 | <a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C1R</a>   | 1555 |
| 7.94.2.19 | <a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C3CMR</a> | 1556 |
| 7.94.2.20 | <a href="#">nppiMinMaxIndxGetBufferHostSize_16u_C3CR</a>  | 1556 |
| 7.94.2.21 | <a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C1MR</a>  | 1556 |
| 7.94.2.22 | <a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C1R</a>   | 1556 |
| 7.94.2.23 | <a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C3CMR</a> | 1557 |
| 7.94.2.24 | <a href="#">nppiMinMaxIndxGetBufferHostSize_32f_C3CR</a>  | 1557 |
| 7.94.2.25 | <a href="#">nppiMinMaxIndxGetBufferHostSize_8s_C1MR</a>   | 1557 |
| 7.94.2.26 | <a href="#">nppiMinMaxIndxGetBufferHostSize_8s_C1R</a>    | 1558 |

|           |   |      |
|-----------|---|------|
| 7.94.2.27 | <a href="#">nppiMinMaxIdxGetBufferHostSize_8s_C3CMR</a> | 1558 |
| 7.94.2.28 | <a href="#">nppiMinMaxIdxGetBufferHostSize_8s_C3CR</a>  | 1558 |
| 7.94.2.29 | <a href="#">nppiMinMaxIdxGetBufferHostSize_8u_C1MR</a>  | 1558 |
| 7.94.2.30 | <a href="#">nppiMinMaxIdxGetBufferHostSize_8u_C1R</a>   | 1559 |
| 7.94.2.31 | <a href="#">nppiMinMaxIdxGetBufferHostSize_8u_C3CMR</a> | 1559 |
| 7.94.2.32 | <a href="#">nppiMinMaxIdxGetBufferHostSize_8u_C3CR</a>  | 1559 |
| 7.95      | <a href="#">Mean</a>                                    | 1560 |
| 7.95.1    | <a href="#">Detailed Description</a>                    | 1563 |
| 7.95.2    | <a href="#">Function Documentation</a>                  | 1564 |
| 7.95.2.1  | <a href="#">nppiMean_16s_AC4R</a>                       | 1564 |
| 7.95.2.2  | <a href="#">nppiMean_16s_C1R</a>                        | 1564 |
| 7.95.2.3  | <a href="#">nppiMean_16s_C3R</a>                        | 1564 |
| 7.95.2.4  | <a href="#">nppiMean_16s_C4R</a>                        | 1565 |
| 7.95.2.5  | <a href="#">nppiMean_16u_AC4R</a>                       | 1565 |
| 7.95.2.6  | <a href="#">nppiMean_16u_C1MR</a>                       | 1565 |
| 7.95.2.7  | <a href="#">nppiMean_16u_C1R</a>                        | 1566 |
| 7.95.2.8  | <a href="#">nppiMean_16u_C3CMR</a>                      | 1566 |
| 7.95.2.9  | <a href="#">nppiMean_16u_C3R</a>                        | 1567 |
| 7.95.2.10 | <a href="#">nppiMean_16u_C4R</a>                        | 1567 |
| 7.95.2.11 | <a href="#">nppiMean_32f_AC4R</a>                       | 1567 |
| 7.95.2.12 | <a href="#">nppiMean_32f_C1MR</a>                       | 1568 |
| 7.95.2.13 | <a href="#">nppiMean_32f_C1R</a>                        | 1568 |
| 7.95.2.14 | <a href="#">nppiMean_32f_C3CMR</a>                      | 1569 |
| 7.95.2.15 | <a href="#">nppiMean_32f_C3R</a>                        | 1569 |
| 7.95.2.16 | <a href="#">nppiMean_32f_C4R</a>                        | 1569 |
| 7.95.2.17 | <a href="#">nppiMean_8s_C1MR</a>                        | 1570 |
| 7.95.2.18 | <a href="#">nppiMean_8s_C3CMR</a>                       | 1570 |
| 7.95.2.19 | <a href="#">nppiMean_8u_AC4R</a>                        | 1571 |
| 7.95.2.20 | <a href="#">nppiMean_8u_C1MR</a>                        | 1571 |
| 7.95.2.21 | <a href="#">nppiMean_8u_C1R</a>                         | 1572 |
| 7.95.2.22 | <a href="#">nppiMean_8u_C3CMR</a>                       | 1572 |
| 7.95.2.23 | <a href="#">nppiMean_8u_C3R</a>                         | 1572 |
| 7.95.2.24 | <a href="#">nppiMean_8u_C4R</a>                         | 1573 |
| 7.95.2.25 | <a href="#">nppiMeanGetBufferHostSize_16s_AC4R</a>      | 1573 |
| 7.95.2.26 | <a href="#">nppiMeanGetBufferHostSize_16s_C1R</a>       | 1573 |
| 7.95.2.27 | <a href="#">nppiMeanGetBufferHostSize_16s_C3R</a>       | 1574 |

|           |   |      |
|-----------|---|------|
| 7.95.2.28 | <a href="#">nppiMeanGetBufferHostSize_16s_C4R</a>   | 1574 |
| 7.95.2.29 | <a href="#">nppiMeanGetBufferHostSize_16u_AC4R</a>  | 1574 |
| 7.95.2.30 | <a href="#">nppiMeanGetBufferHostSize_16u_C1MR</a>  | 1575 |
| 7.95.2.31 | <a href="#">nppiMeanGetBufferHostSize_16u_C1R</a>   | 1575 |
| 7.95.2.32 | <a href="#">nppiMeanGetBufferHostSize_16u_C3CMR</a> | 1575 |
| 7.95.2.33 | <a href="#">nppiMeanGetBufferHostSize_16u_C3R</a>   | 1575 |
| 7.95.2.34 | <a href="#">nppiMeanGetBufferHostSize_16u_C4R</a>   | 1576 |
| 7.95.2.35 | <a href="#">nppiMeanGetBufferHostSize_32f_AC4R</a>  | 1576 |
| 7.95.2.36 | <a href="#">nppiMeanGetBufferHostSize_32f_C1MR</a>  | 1576 |
| 7.95.2.37 | <a href="#">nppiMeanGetBufferHostSize_32f_C1R</a>   | 1577 |
| 7.95.2.38 | <a href="#">nppiMeanGetBufferHostSize_32f_C3CMR</a> | 1577 |
| 7.95.2.39 | <a href="#">nppiMeanGetBufferHostSize_32f_C3R</a>   | 1577 |
| 7.95.2.40 | <a href="#">nppiMeanGetBufferHostSize_32f_C4R</a>   | 1577 |
| 7.95.2.41 | <a href="#">nppiMeanGetBufferHostSize_8s_C1MR</a>   | 1578 |
| 7.95.2.42 | <a href="#">nppiMeanGetBufferHostSize_8s_C3CMR</a>  | 1578 |
| 7.95.2.43 | <a href="#">nppiMeanGetBufferHostSize_8u_AC4R</a>   | 1578 |
| 7.95.2.44 | <a href="#">nppiMeanGetBufferHostSize_8u_C1MR</a>   | 1579 |
| 7.95.2.45 | <a href="#">nppiMeanGetBufferHostSize_8u_C1R</a>    | 1579 |
| 7.95.2.46 | <a href="#">nppiMeanGetBufferHostSize_8u_C3CMR</a>  | 1579 |
| 7.95.2.47 | <a href="#">nppiMeanGetBufferHostSize_8u_C3R</a>    | 1579 |
| 7.95.2.48 | <a href="#">nppiMeanGetBufferHostSize_8u_C4R</a>    | 1580 |
| 7.96      | <a href="#">Mean_StdDev</a>                         | 1581 |
| 7.96.1    | <a href="#">Detailed Description</a>                | 1584 |
| 7.96.2    | <a href="#">Function Documentation</a>              | 1584 |
| 7.96.2.1  | <a href="#">nppiMean_StdDev_16u_C1MR</a>            | 1584 |
| 7.96.2.2  | <a href="#">nppiMean_StdDev_16u_C1R</a>             | 1585 |
| 7.96.2.3  | <a href="#">nppiMean_StdDev_16u_C3CMR</a>           | 1585 |
| 7.96.2.4  | <a href="#">nppiMean_StdDev_16u_C3CR</a>            | 1586 |
| 7.96.2.5  | <a href="#">nppiMean_StdDev_32f_C1MR</a>            | 1586 |
| 7.96.2.6  | <a href="#">nppiMean_StdDev_32f_C1R</a>             | 1587 |
| 7.96.2.7  | <a href="#">nppiMean_StdDev_32f_C3CMR</a>           | 1587 |
| 7.96.2.8  | <a href="#">nppiMean_StdDev_32f_C3CR</a>            | 1588 |
| 7.96.2.9  | <a href="#">nppiMean_StdDev_8s_C1MR</a>             | 1588 |
| 7.96.2.10 | <a href="#">nppiMean_StdDev_8s_C1R</a>              | 1589 |
| 7.96.2.11 | <a href="#">nppiMean_StdDev_8s_C3CMR</a>            | 1589 |
| 7.96.2.12 | <a href="#">nppiMean_StdDev_8s_C3CR</a>             | 1590 |

|           |   |      |
|-----------|---|------|
| 7.96.2.13 | <a href="#">nppiMean_StdDev_8u_C1MR</a>                   | 1590 |
| 7.96.2.14 | <a href="#">nppiMean_StdDev_8u_C1R</a>                    | 1591 |
| 7.96.2.15 | <a href="#">nppiMean_StdDev_8u_C3CMR</a>                  | 1591 |
| 7.96.2.16 | <a href="#">nppiMean_StdDev_8u_C3CR</a>                   | 1592 |
| 7.96.2.17 | <a href="#">nppiMeanStdDevGetBufferHostSize_16u_C1MR</a>  | 1592 |
| 7.96.2.18 | <a href="#">nppiMeanStdDevGetBufferHostSize_16u_C1R</a>   | 1592 |
| 7.96.2.19 | <a href="#">nppiMeanStdDevGetBufferHostSize_16u_C3CMR</a> | 1593 |
| 7.96.2.20 | <a href="#">nppiMeanStdDevGetBufferHostSize_16u_C3CR</a>  | 1593 |
| 7.96.2.21 | <a href="#">nppiMeanStdDevGetBufferHostSize_32f_C1MR</a>  | 1593 |
| 7.96.2.22 | <a href="#">nppiMeanStdDevGetBufferHostSize_32f_C1R</a>   | 1593 |
| 7.96.2.23 | <a href="#">nppiMeanStdDevGetBufferHostSize_32f_C3CMR</a> | 1594 |
| 7.96.2.24 | <a href="#">nppiMeanStdDevGetBufferHostSize_32f_C3CR</a>  | 1594 |
| 7.96.2.25 | <a href="#">nppiMeanStdDevGetBufferHostSize_8s_C1MR</a>   | 1594 |
| 7.96.2.26 | <a href="#">nppiMeanStdDevGetBufferHostSize_8s_C1R</a>    | 1595 |
| 7.96.2.27 | <a href="#">nppiMeanStdDevGetBufferHostSize_8s_C3CMR</a>  | 1595 |
| 7.96.2.28 | <a href="#">nppiMeanStdDevGetBufferHostSize_8s_C3CR</a>   | 1595 |
| 7.96.2.29 | <a href="#">nppiMeanStdDevGetBufferHostSize_8u_C1MR</a>   | 1595 |
| 7.96.2.30 | <a href="#">nppiMeanStdDevGetBufferHostSize_8u_C1R</a>    | 1596 |
| 7.96.2.31 | <a href="#">nppiMeanStdDevGetBufferHostSize_8u_C3CMR</a>  | 1596 |
| 7.96.2.32 | <a href="#">nppiMeanStdDevGetBufferHostSize_8u_C3CR</a>   | 1596 |
| 7.97      | <a href="#">Image Norms</a>                               | 1597 |
| 7.97.1    | <a href="#">Detailed Description</a>                      | 1597 |
| 7.98      | <a href="#">Norm_Inf</a>                                  | 1599 |
| 7.98.1    | <a href="#">Detailed Description</a>                      | 1603 |
| 7.98.2    | <a href="#">Function Documentation</a>                    | 1603 |
| 7.98.2.1  | <a href="#">nppiNorm_Inf_16s_AC4R</a>                     | 1603 |
| 7.98.2.2  | <a href="#">nppiNorm_Inf_16s_C1R</a>                      | 1603 |
| 7.98.2.3  | <a href="#">nppiNorm_Inf_16s_C3R</a>                      | 1603 |
| 7.98.2.4  | <a href="#">nppiNorm_Inf_16s_C4R</a>                      | 1604 |
| 7.98.2.5  | <a href="#">nppiNorm_Inf_16u_AC4R</a>                     | 1604 |
| 7.98.2.6  | <a href="#">nppiNorm_Inf_16u_C1MR</a>                     | 1605 |
| 7.98.2.7  | <a href="#">nppiNorm_Inf_16u_C1R</a>                      | 1605 |
| 7.98.2.8  | <a href="#">nppiNorm_Inf_16u_C3CMR</a>                    | 1605 |
| 7.98.2.9  | <a href="#">nppiNorm_Inf_16u_C3R</a>                      | 1606 |
| 7.98.2.10 | <a href="#">nppiNorm_Inf_16u_C4R</a>                      | 1606 |
| 7.98.2.11 | <a href="#">nppiNorm_Inf_32f_AC4R</a>                     | 1607 |

|  |      |
|--|------|
| 7.98.2.12 nppiNorm_Inf_32f_C1MR . . . . .                  | 1607 |
| 7.98.2.13 nppiNorm_Inf_32f_C1R . . . . .                   | 1607 |
| 7.98.2.14 nppiNorm_Inf_32f_C3CMR . . . . .                 | 1608 |
| 7.98.2.15 nppiNorm_Inf_32f_C3R . . . . .                   | 1608 |
| 7.98.2.16 nppiNorm_Inf_32f_C4R . . . . .                   | 1609 |
| 7.98.2.17 nppiNorm_Inf_32s_C1R . . . . .                   | 1609 |
| 7.98.2.18 nppiNorm_Inf_8s_C1MR . . . . .                   | 1609 |
| 7.98.2.19 nppiNorm_Inf_8s_C3CMR . . . . .                  | 1610 |
| 7.98.2.20 nppiNorm_Inf_8u_AC4R . . . . .                   | 1610 |
| 7.98.2.21 nppiNorm_Inf_8u_C1MR . . . . .                   | 1611 |
| 7.98.2.22 nppiNorm_Inf_8u_C1R . . . . .                    | 1611 |
| 7.98.2.23 nppiNorm_Inf_8u_C3CMR . . . . .                  | 1611 |
| 7.98.2.24 nppiNorm_Inf_8u_C3R . . . . .                    | 1612 |
| 7.98.2.25 nppiNorm_Inf_8u_C4R . . . . .                    | 1612 |
| 7.98.2.26 nppiNormInfGetBufferHostSize_16s_AC4R . . . . .  | 1613 |
| 7.98.2.27 nppiNormInfGetBufferHostSize_16s_C1R . . . . .   | 1613 |
| 7.98.2.28 nppiNormInfGetBufferHostSize_16s_C3R . . . . .   | 1613 |
| 7.98.2.29 nppiNormInfGetBufferHostSize_16s_C4R . . . . .   | 1613 |
| 7.98.2.30 nppiNormInfGetBufferHostSize_16u_AC4R . . . . .  | 1614 |
| 7.98.2.31 nppiNormInfGetBufferHostSize_16u_C1MR . . . . .  | 1614 |
| 7.98.2.32 nppiNormInfGetBufferHostSize_16u_C1R . . . . .   | 1614 |
| 7.98.2.33 nppiNormInfGetBufferHostSize_16u_C3CMR . . . . . | 1615 |
| 7.98.2.34 nppiNormInfGetBufferHostSize_16u_C3R . . . . .   | 1615 |
| 7.98.2.35 nppiNormInfGetBufferHostSize_16u_C4R . . . . .   | 1615 |
| 7.98.2.36 nppiNormInfGetBufferHostSize_32f_AC4R . . . . .  | 1615 |
| 7.98.2.37 nppiNormInfGetBufferHostSize_32f_C1MR . . . . .  | 1616 |
| 7.98.2.38 nppiNormInfGetBufferHostSize_32f_C1R . . . . .   | 1616 |
| 7.98.2.39 nppiNormInfGetBufferHostSize_32f_C3CMR . . . . . | 1616 |
| 7.98.2.40 nppiNormInfGetBufferHostSize_32f_C3R . . . . .   | 1617 |
| 7.98.2.41 nppiNormInfGetBufferHostSize_32f_C4R . . . . .   | 1617 |
| 7.98.2.42 nppiNormInfGetBufferHostSize_32s_C1R . . . . .   | 1617 |
| 7.98.2.43 nppiNormInfGetBufferHostSize_8s_C1MR . . . . .   | 1617 |
| 7.98.2.44 nppiNormInfGetBufferHostSize_8s_C3CMR . . . . .  | 1618 |
| 7.98.2.45 nppiNormInfGetBufferHostSize_8u_AC4R . . . . .   | 1618 |
| 7.98.2.46 nppiNormInfGetBufferHostSize_8u_C1MR . . . . .   | 1618 |
| 7.98.2.47 nppiNormInfGetBufferHostSize_8u_C1R . . . . .    | 1619 |

|           |   |      |
|-----------|---|------|
| 7.98.2.48 | <a href="#">nppiNormInfGetBufferHostSize_8u_C3CMR</a> | 1619 |
| 7.98.2.49 | <a href="#">nppiNormInfGetBufferHostSize_8u_C3R</a>   | 1619 |
| 7.98.2.50 | <a href="#">nppiNormInfGetBufferHostSize_8u_C4R</a>   | 1619 |
| 7.99      | <a href="#">Norm_L1</a>                               | 1621 |
| 7.99.1    | <a href="#">Detailed Description</a>                  | 1624 |
| 7.99.2    | <a href="#">Function Documentation</a>                | 1625 |
| 7.99.2.1  | <a href="#">nppiNorm_L1_16s_AC4R</a>                  | 1625 |
| 7.99.2.2  | <a href="#">nppiNorm_L1_16s_C1R</a>                   | 1625 |
| 7.99.2.3  | <a href="#">nppiNorm_L1_16s_C3R</a>                   | 1625 |
| 7.99.2.4  | <a href="#">nppiNorm_L1_16s_C4R</a>                   | 1626 |
| 7.99.2.5  | <a href="#">nppiNorm_L1_16u_AC4R</a>                  | 1626 |
| 7.99.2.6  | <a href="#">nppiNorm_L1_16u_C1MR</a>                  | 1626 |
| 7.99.2.7  | <a href="#">nppiNorm_L1_16u_C1R</a>                   | 1627 |
| 7.99.2.8  | <a href="#">nppiNorm_L1_16u_C3CMR</a>                 | 1627 |
| 7.99.2.9  | <a href="#">nppiNorm_L1_16u_C3R</a>                   | 1628 |
| 7.99.2.10 | <a href="#">nppiNorm_L1_16u_C4R</a>                   | 1628 |
| 7.99.2.11 | <a href="#">nppiNorm_L1_32f_AC4R</a>                  | 1628 |
| 7.99.2.12 | <a href="#">nppiNorm_L1_32f_C1MR</a>                  | 1629 |
| 7.99.2.13 | <a href="#">nppiNorm_L1_32f_C1R</a>                   | 1629 |
| 7.99.2.14 | <a href="#">nppiNorm_L1_32f_C3CMR</a>                 | 1630 |
| 7.99.2.15 | <a href="#">nppiNorm_L1_32f_C3R</a>                   | 1630 |
| 7.99.2.16 | <a href="#">nppiNorm_L1_32f_C4R</a>                   | 1630 |
| 7.99.2.17 | <a href="#">nppiNorm_L1_8s_C1MR</a>                   | 1631 |
| 7.99.2.18 | <a href="#">nppiNorm_L1_8s_C3CMR</a>                  | 1631 |
| 7.99.2.19 | <a href="#">nppiNorm_L1_8u_AC4R</a>                   | 1632 |
| 7.99.2.20 | <a href="#">nppiNorm_L1_8u_C1MR</a>                   | 1632 |
| 7.99.2.21 | <a href="#">nppiNorm_L1_8u_C1R</a>                    | 1632 |
| 7.99.2.22 | <a href="#">nppiNorm_L1_8u_C3CMR</a>                  | 1633 |
| 7.99.2.23 | <a href="#">nppiNorm_L1_8u_C3R</a>                    | 1633 |
| 7.99.2.24 | <a href="#">nppiNorm_L1_8u_C4R</a>                    | 1634 |
| 7.99.2.25 | <a href="#">nppiNormL1GetBufferHostSize_16s_AC4R</a>  | 1634 |
| 7.99.2.26 | <a href="#">nppiNormL1GetBufferHostSize_16s_C1R</a>   | 1634 |
| 7.99.2.27 | <a href="#">nppiNormL1GetBufferHostSize_16s_C3R</a>   | 1635 |
| 7.99.2.28 | <a href="#">nppiNormL1GetBufferHostSize_16s_C4R</a>   | 1635 |
| 7.99.2.29 | <a href="#">nppiNormL1GetBufferHostSize_16u_AC4R</a>  | 1635 |
| 7.99.2.30 | <a href="#">nppiNormL1GetBufferHostSize_16u_C1MR</a>  | 1635 |



|   |      |
|---|------|
| 7.99.2.31 nppiNormL1GetBufferHostSize_16u_C1R . . . . .   | 1636 |
| 7.99.2.32 nppiNormL1GetBufferHostSize_16u_C3CMR . . . . . | 1636 |
| 7.99.2.33 nppiNormL1GetBufferHostSize_16u_C3R . . . . .   | 1636 |
| 7.99.2.34 nppiNormL1GetBufferHostSize_16u_C4R . . . . .   | 1637 |
| 7.99.2.35 nppiNormL1GetBufferHostSize_32f_AC4R . . . . .  | 1637 |
| 7.99.2.36 nppiNormL1GetBufferHostSize_32f_C1MR . . . . .  | 1637 |
| 7.99.2.37 nppiNormL1GetBufferHostSize_32f_C1R . . . . .   | 1637 |
| 7.99.2.38 nppiNormL1GetBufferHostSize_32f_C3CMR . . . . . | 1638 |
| 7.99.2.39 nppiNormL1GetBufferHostSize_32f_C3R . . . . .   | 1638 |
| 7.99.2.40 nppiNormL1GetBufferHostSize_32f_C4R . . . . .   | 1638 |
| 7.99.2.41 nppiNormL1GetBufferHostSize_8s_C1MR . . . . .   | 1639 |
| 7.99.2.42 nppiNormL1GetBufferHostSize_8s_C3CMR . . . . .  | 1639 |
| 7.99.2.43 nppiNormL1GetBufferHostSize_8u_AC4R . . . . .   | 1639 |
| 7.99.2.44 nppiNormL1GetBufferHostSize_8u_C1MR . . . . .   | 1639 |
| 7.99.2.45 nppiNormL1GetBufferHostSize_8u_C1R . . . . .    | 1640 |
| 7.99.2.46 nppiNormL1GetBufferHostSize_8u_C3CMR . . . . .  | 1640 |
| 7.99.2.47 nppiNormL1GetBufferHostSize_8u_C3R . . . . .    | 1640 |
| 7.99.2.48 nppiNormL1GetBufferHostSize_8u_C4R . . . . .    | 1641 |
| 7.100Norm_L2 . . . . .                                    | 1642 |
| 7.100.1 Detailed Description . . . . .                    | 1645 |
| 7.100.2 Function Documentation . . . . .                  | 1646 |
| 7.100.2.1 nppiNorm_L2_16s_AC4R . . . . .                  | 1646 |
| 7.100.2.2 nppiNorm_L2_16s_C1R . . . . .                   | 1646 |
| 7.100.2.3 nppiNorm_L2_16s_C3R . . . . .                   | 1646 |
| 7.100.2.4 nppiNorm_L2_16s_C4R . . . . .                   | 1647 |
| 7.100.2.5 nppiNorm_L2_16u_AC4R . . . . .                  | 1647 |
| 7.100.2.6 nppiNorm_L2_16u_C1MR . . . . .                  | 1647 |
| 7.100.2.7 nppiNorm_L2_16u_C1R . . . . .                   | 1648 |
| 7.100.2.8 nppiNorm_L2_16u_C3CMR . . . . .                 | 1648 |
| 7.100.2.9 nppiNorm_L2_16u_C3R . . . . .                   | 1649 |
| 7.100.2.10 nppiNorm_L2_16u_C4R . . . . .                  | 1649 |
| 7.100.2.11 nppiNorm_L2_32f_AC4R . . . . .                 | 1649 |
| 7.100.2.12 nppiNorm_L2_32f_C1MR . . . . .                 | 1650 |
| 7.100.2.13 nppiNorm_L2_32f_C1R . . . . .                  | 1650 |
| 7.100.2.14 nppiNorm_L2_32f_C3CMR . . . . .                | 1651 |
| 7.100.2.15 nppiNorm_L2_32f_C3R . . . . .                  | 1651 |

|            |  |      |
|------------|--|------|
| 7.100.2.16 | ppiNorm_L2_32f_C4R . . . . .                   | 1651 |
| 7.100.2.17 | ppiNorm_L2_8s_C1MR . . . . .                   | 1652 |
| 7.100.2.18 | ppiNorm_L2_8s_C3CMR . . . . .                  | 1652 |
| 7.100.2.19 | ppiNorm_L2_8u_AC4R . . . . .                   | 1653 |
| 7.100.2.20 | ppiNorm_L2_8u_C1MR . . . . .                   | 1653 |
| 7.100.2.21 | ppiNorm_L2_8u_C1R . . . . .                    | 1653 |
| 7.100.2.22 | ppiNorm_L2_8u_C3CMR . . . . .                  | 1654 |
| 7.100.2.23 | ppiNorm_L2_8u_C3R . . . . .                    | 1654 |
| 7.100.2.24 | ppiNorm_L2_8u_C4R . . . . .                    | 1655 |
| 7.100.2.25 | ppiNormL2GetBufferHostSize_16s_AC4R . . . . .  | 1655 |
| 7.100.2.26 | ppiNormL2GetBufferHostSize_16s_C1R . . . . .   | 1655 |
| 7.100.2.27 | ppiNormL2GetBufferHostSize_16s_C3R . . . . .   | 1656 |
| 7.100.2.28 | ppiNormL2GetBufferHostSize_16s_C4R . . . . .   | 1656 |
| 7.100.2.29 | ppiNormL2GetBufferHostSize_16u_AC4R . . . . .  | 1656 |
| 7.100.2.30 | ppiNormL2GetBufferHostSize_16u_C1MR . . . . .  | 1656 |
| 7.100.2.31 | ppiNormL2GetBufferHostSize_16u_C1R . . . . .   | 1657 |
| 7.100.2.32 | ppiNormL2GetBufferHostSize_16u_C3CMR . . . . . | 1657 |
| 7.100.2.33 | ppiNormL2GetBufferHostSize_16u_C3R . . . . .   | 1657 |
| 7.100.2.34 | ppiNormL2GetBufferHostSize_16u_C4R . . . . .   | 1658 |
| 7.100.2.35 | ppiNormL2GetBufferHostSize_32f_AC4R . . . . .  | 1658 |
| 7.100.2.36 | ppiNormL2GetBufferHostSize_32f_C1MR . . . . .  | 1658 |
| 7.100.2.37 | ppiNormL2GetBufferHostSize_32f_C1R . . . . .   | 1658 |
| 7.100.2.38 | ppiNormL2GetBufferHostSize_32f_C3CMR . . . . . | 1659 |
| 7.100.2.39 | ppiNormL2GetBufferHostSize_32f_C3R . . . . .   | 1659 |
| 7.100.2.40 | ppiNormL2GetBufferHostSize_32f_C4R . . . . .   | 1659 |
| 7.100.2.41 | ppiNormL2GetBufferHostSize_8s_C1MR . . . . .   | 1660 |
| 7.100.2.42 | ppiNormL2GetBufferHostSize_8s_C3CMR . . . . .  | 1660 |
| 7.100.2.43 | ppiNormL2GetBufferHostSize_8u_AC4R . . . . .   | 1660 |
| 7.100.2.44 | ppiNormL2GetBufferHostSize_8u_C1MR . . . . .   | 1660 |
| 7.100.2.45 | ppiNormL2GetBufferHostSize_8u_C1R . . . . .    | 1661 |
| 7.100.2.46 | ppiNormL2GetBufferHostSize_8u_C3CMR . . . . .  | 1661 |
| 7.100.2.47 | ppiNormL2GetBufferHostSize_8u_C3R . . . . .    | 1661 |
| 7.100.2.48 | ppiNormL2GetBufferHostSize_8u_C4R . . . . .    | 1662 |
| 7.101      | NormDiff_Inf . . . . .                         | 1663 |
| 7.101.1    | Detailed Description . . . . .                 | 1667 |
| 7.101.2    | Function Documentation . . . . .               | 1667 |

|   |      |
|---|------|
| 7.101.2.1 nppiNormDiff_Inf_16s_AC4R . . . . .                   | 1667 |
| 7.101.2.2 nppiNormDiff_Inf_16s_C1R . . . . .                    | 1668 |
| 7.101.2.3 nppiNormDiff_Inf_16s_C3R . . . . .                    | 1668 |
| 7.101.2.4 nppiNormDiff_Inf_16s_C4R . . . . .                    | 1668 |
| 7.101.2.5 nppiNormDiff_Inf_16u_AC4R . . . . .                   | 1669 |
| 7.101.2.6 nppiNormDiff_Inf_16u_C1MR . . . . .                   | 1669 |
| 7.101.2.7 nppiNormDiff_Inf_16u_C1R . . . . .                    | 1670 |
| 7.101.2.8 nppiNormDiff_Inf_16u_C3CMR . . . . .                  | 1670 |
| 7.101.2.9 nppiNormDiff_Inf_16u_C3R . . . . .                    | 1671 |
| 7.101.2.10 nppiNormDiff_Inf_16u_C4R . . . . .                   | 1671 |
| 7.101.2.11 nppiNormDiff_Inf_32f_AC4R . . . . .                  | 1672 |
| 7.101.2.12 nppiNormDiff_Inf_32f_C1MR . . . . .                  | 1672 |
| 7.101.2.13 nppiNormDiff_Inf_32f_C1R . . . . .                   | 1673 |
| 7.101.2.14 nppiNormDiff_Inf_32f_C3CMR . . . . .                 | 1673 |
| 7.101.2.15 nppiNormDiff_Inf_32f_C3R . . . . .                   | 1674 |
| 7.101.2.16 nppiNormDiff_Inf_32f_C4R . . . . .                   | 1674 |
| 7.101.2.17 nppiNormDiff_Inf_8s_C1MR . . . . .                   | 1674 |
| 7.101.2.18 nppiNormDiff_Inf_8s_C3CMR . . . . .                  | 1675 |
| 7.101.2.19 nppiNormDiff_Inf_8u_AC4R . . . . .                   | 1675 |
| 7.101.2.20 nppiNormDiff_Inf_8u_C1MR . . . . .                   | 1676 |
| 7.101.2.21 nppiNormDiff_Inf_8u_C1R . . . . .                    | 1676 |
| 7.101.2.22 nppiNormDiff_Inf_8u_C3CMR . . . . .                  | 1677 |
| 7.101.2.23 nppiNormDiff_Inf_8u_C3R . . . . .                    | 1677 |
| 7.101.2.24 nppiNormDiff_Inf_8u_C4R . . . . .                    | 1678 |
| 7.101.2.25 nppiNormDiffInfGetBufferHostSize_16s_AC4R . . . . .  | 1678 |
| 7.101.2.26 nppiNormDiffInfGetBufferHostSize_16s_C1R . . . . .   | 1679 |
| 7.101.2.27 nppiNormDiffInfGetBufferHostSize_16s_C3R . . . . .   | 1679 |
| 7.101.2.28 nppiNormDiffInfGetBufferHostSize_16s_C4R . . . . .   | 1679 |
| 7.101.2.29 nppiNormDiffInfGetBufferHostSize_16u_AC4R . . . . .  | 1679 |
| 7.101.2.30 nppiNormDiffInfGetBufferHostSize_16u_C1MR . . . . .  | 1680 |
| 7.101.2.31 nppiNormDiffInfGetBufferHostSize_16u_C1R . . . . .   | 1680 |
| 7.101.2.32 nppiNormDiffInfGetBufferHostSize_16u_C3CMR . . . . . | 1680 |
| 7.101.2.33 nppiNormDiffInfGetBufferHostSize_16u_C3R . . . . .   | 1681 |
| 7.101.2.34 nppiNormDiffInfGetBufferHostSize_16u_C4R . . . . .   | 1681 |
| 7.101.2.35 nppiNormDiffInfGetBufferHostSize_32f_AC4R . . . . .  | 1681 |
| 7.101.2.36 nppiNormDiffInfGetBufferHostSize_32f_C1MR . . . . .  | 1681 |

|            |   |      |
|------------|---|------|
| 7.101.2.37 | npapiNormDiffInfGetBufferHostSize_32f_C1R . . . . .   | 1682 |
| 7.101.2.38 | npapiNormDiffInfGetBufferHostSize_32f_C3CMR . . . . . | 1682 |
| 7.101.2.39 | npapiNormDiffInfGetBufferHostSize_32f_C3R . . . . .   | 1682 |
| 7.101.2.40 | npapiNormDiffInfGetBufferHostSize_32f_C4R . . . . .   | 1683 |
| 7.101.2.41 | npapiNormDiffInfGetBufferHostSize_8s_C1MR . . . . .   | 1683 |
| 7.101.2.42 | npapiNormDiffInfGetBufferHostSize_8s_C3CMR . . . . .  | 1683 |
| 7.101.2.43 | npapiNormDiffInfGetBufferHostSize_8u_AC4R . . . . .   | 1683 |
| 7.101.2.44 | npapiNormDiffInfGetBufferHostSize_8u_C1MR . . . . .   | 1684 |
| 7.101.2.45 | npapiNormDiffInfGetBufferHostSize_8u_C1R . . . . .    | 1684 |
| 7.101.2.46 | npapiNormDiffInfGetBufferHostSize_8u_C3CMR . . . . .  | 1684 |
| 7.101.2.47 | npapiNormDiffInfGetBufferHostSize_8u_C3R . . . . .    | 1685 |
| 7.101.2.48 | npapiNormDiffInfGetBufferHostSize_8u_C4R . . . . .    | 1685 |
| 7.102      | NormDiff_L1 . . . . .                                 | 1686 |
| 7.102.1    | Detailed Description . . . . .                        | 1690 |
| 7.102.2    | Function Documentation . . . . .                      | 1690 |
| 7.102.2.1  | npapiNormDiff_L1_16s_AC4R . . . . .                   | 1690 |
| 7.102.2.2  | npapiNormDiff_L1_16s_C1R . . . . .                    | 1690 |
| 7.102.2.3  | npapiNormDiff_L1_16s_C3R . . . . .                    | 1691 |
| 7.102.2.4  | npapiNormDiff_L1_16s_C4R . . . . .                    | 1691 |
| 7.102.2.5  | npapiNormDiff_L1_16u_AC4R . . . . .                   | 1692 |
| 7.102.2.6  | npapiNormDiff_L1_16u_C1MR . . . . .                   | 1692 |
| 7.102.2.7  | npapiNormDiff_L1_16u_C1R . . . . .                    | 1693 |
| 7.102.2.8  | npapiNormDiff_L1_16u_C3CMR . . . . .                  | 1693 |
| 7.102.2.9  | npapiNormDiff_L1_16u_C3R . . . . .                    | 1694 |
| 7.102.2.10 | npapiNormDiff_L1_16u_C4R . . . . .                    | 1694 |
| 7.102.2.11 | npapiNormDiff_L1_32f_AC4R . . . . .                   | 1694 |
| 7.102.2.12 | npapiNormDiff_L1_32f_C1MR . . . . .                   | 1695 |
| 7.102.2.13 | npapiNormDiff_L1_32f_C1R . . . . .                    | 1695 |
| 7.102.2.14 | npapiNormDiff_L1_32f_C3CMR . . . . .                  | 1696 |
| 7.102.2.15 | npapiNormDiff_L1_32f_C3R . . . . .                    | 1696 |
| 7.102.2.16 | npapiNormDiff_L1_32f_C4R . . . . .                    | 1697 |
| 7.102.2.17 | npapiNormDiff_L1_8s_C1MR . . . . .                    | 1697 |
| 7.102.2.18 | npapiNormDiff_L1_8s_C3CMR . . . . .                   | 1698 |
| 7.102.2.19 | npapiNormDiff_L1_8u_AC4R . . . . .                    | 1698 |
| 7.102.2.20 | npapiNormDiff_L1_8u_C1MR . . . . .                    | 1699 |
| 7.102.2.21 | npapiNormDiff_L1_8u_C1R . . . . .                     | 1699 |

|            |   |      |
|------------|---|------|
| 7.102.2.22 | nppiNormDiff_L1_8u_C3CMR . . . . .                  | 1700 |
| 7.102.2.23 | nppiNormDiff_L1_8u_C3R . . . . .                    | 1700 |
| 7.102.2.24 | nppiNormDiff_L1_8u_C4R . . . . .                    | 1701 |
| 7.102.2.25 | nppiNormDiffL1GetBufferHostSize_16s_AC4R . . . . .  | 1701 |
| 7.102.2.26 | nppiNormDiffL1GetBufferHostSize_16s_C1R . . . . .   | 1701 |
| 7.102.2.27 | nppiNormDiffL1GetBufferHostSize_16s_C3R . . . . .   | 1702 |
| 7.102.2.28 | nppiNormDiffL1GetBufferHostSize_16s_C4R . . . . .   | 1702 |
| 7.102.2.29 | nppiNormDiffL1GetBufferHostSize_16u_AC4R . . . . .  | 1702 |
| 7.102.2.30 | nppiNormDiffL1GetBufferHostSize_16u_C1MR . . . . .  | 1702 |
| 7.102.2.31 | nppiNormDiffL1GetBufferHostSize_16u_C1R . . . . .   | 1703 |
| 7.102.2.32 | nppiNormDiffL1GetBufferHostSize_16u_C3CMR . . . . . | 1703 |
| 7.102.2.33 | nppiNormDiffL1GetBufferHostSize_16u_C3R . . . . .   | 1703 |
| 7.102.2.34 | nppiNormDiffL1GetBufferHostSize_16u_C4R . . . . .   | 1704 |
| 7.102.2.35 | nppiNormDiffL1GetBufferHostSize_32f_AC4R . . . . .  | 1704 |
| 7.102.2.36 | nppiNormDiffL1GetBufferHostSize_32f_C1MR . . . . .  | 1704 |
| 7.102.2.37 | nppiNormDiffL1GetBufferHostSize_32f_C1R . . . . .   | 1704 |
| 7.102.2.38 | nppiNormDiffL1GetBufferHostSize_32f_C3CMR . . . . . | 1705 |
| 7.102.2.39 | nppiNormDiffL1GetBufferHostSize_32f_C3R . . . . .   | 1705 |
| 7.102.2.40 | nppiNormDiffL1GetBufferHostSize_32f_C4R . . . . .   | 1705 |
| 7.102.2.41 | nppiNormDiffL1GetBufferHostSize_8s_C1MR . . . . .   | 1706 |
| 7.102.2.42 | nppiNormDiffL1GetBufferHostSize_8s_C3CMR . . . . .  | 1706 |
| 7.102.2.43 | nppiNormDiffL1GetBufferHostSize_8u_AC4R . . . . .   | 1706 |
| 7.102.2.44 | nppiNormDiffL1GetBufferHostSize_8u_C1MR . . . . .   | 1706 |
| 7.102.2.45 | nppiNormDiffL1GetBufferHostSize_8u_C1R . . . . .    | 1707 |
| 7.102.2.46 | nppiNormDiffL1GetBufferHostSize_8u_C3CMR . . . . .  | 1707 |
| 7.102.2.47 | nppiNormDiffL1GetBufferHostSize_8u_C3R . . . . .    | 1707 |
| 7.102.2.48 | nppiNormDiffL1GetBufferHostSize_8u_C4R . . . . .    | 1708 |
| 7.103      | NormDiff_L2 . . . . .                               | 1709 |
| 7.103.1    | Detailed Description . . . . .                      | 1713 |
| 7.103.2    | Function Documentation . . . . .                    | 1713 |
| 7.103.2.1  | nppiNormDiff_L2_16s_AC4R . . . . .                  | 1713 |
| 7.103.2.2  | nppiNormDiff_L2_16s_C1R . . . . .                   | 1713 |
| 7.103.2.3  | nppiNormDiff_L2_16s_C3R . . . . .                   | 1714 |
| 7.103.2.4  | nppiNormDiff_L2_16s_C4R . . . . .                   | 1714 |
| 7.103.2.5  | nppiNormDiff_L2_16u_AC4R . . . . .                  | 1715 |
| 7.103.2.6  | nppiNormDiff_L2_16u_C1MR . . . . .                  | 1715 |

|  |      |
|--|------|
| 7.103.2.7 nppiNormDiff_L2_16u_C1R . . . . .                    | 1716 |
| 7.103.2.8 nppiNormDiff_L2_16u_C3CMR . . . . .                  | 1716 |
| 7.103.2.9 nppiNormDiff_L2_16u_C3R . . . . .                    | 1717 |
| 7.103.2.10 nppiNormDiff_L2_16u_C4R . . . . .                   | 1717 |
| 7.103.2.11 nppiNormDiff_L2_32f_AC4R . . . . .                  | 1717 |
| 7.103.2.12 nppiNormDiff_L2_32f_C1MR . . . . .                  | 1718 |
| 7.103.2.13 nppiNormDiff_L2_32f_C1R . . . . .                   | 1718 |
| 7.103.2.14 nppiNormDiff_L2_32f_C3CMR . . . . .                 | 1719 |
| 7.103.2.15 nppiNormDiff_L2_32f_C3R . . . . .                   | 1719 |
| 7.103.2.16 nppiNormDiff_L2_32f_C4R . . . . .                   | 1720 |
| 7.103.2.17 nppiNormDiff_L2_8s_C1MR . . . . .                   | 1720 |
| 7.103.2.18 nppiNormDiff_L2_8s_C3CMR . . . . .                  | 1721 |
| 7.103.2.19 nppiNormDiff_L2_8u_AC4R . . . . .                   | 1721 |
| 7.103.2.20 nppiNormDiff_L2_8u_C1MR . . . . .                   | 1722 |
| 7.103.2.21 nppiNormDiff_L2_8u_C1R . . . . .                    | 1722 |
| 7.103.2.22 nppiNormDiff_L2_8u_C3CMR . . . . .                  | 1723 |
| 7.103.2.23 nppiNormDiff_L2_8u_C3R . . . . .                    | 1723 |
| 7.103.2.24 nppiNormDiff_L2_8u_C4R . . . . .                    | 1724 |
| 7.103.2.25 nppiNormDiffL2GetBufferHostSize_16s_AC4R . . . . .  | 1724 |
| 7.103.2.26 nppiNormDiffL2GetBufferHostSize_16s_C1R . . . . .   | 1724 |
| 7.103.2.27 nppiNormDiffL2GetBufferHostSize_16s_C3R . . . . .   | 1725 |
| 7.103.2.28 nppiNormDiffL2GetBufferHostSize_16s_C4R . . . . .   | 1725 |
| 7.103.2.29 nppiNormDiffL2GetBufferHostSize_16u_AC4R . . . . .  | 1725 |
| 7.103.2.30 nppiNormDiffL2GetBufferHostSize_16u_C1MR . . . . .  | 1725 |
| 7.103.2.31 nppiNormDiffL2GetBufferHostSize_16u_C1R . . . . .   | 1726 |
| 7.103.2.32 nppiNormDiffL2GetBufferHostSize_16u_C3CMR . . . . . | 1726 |
| 7.103.2.33 nppiNormDiffL2GetBufferHostSize_16u_C3R . . . . .   | 1726 |
| 7.103.2.34 nppiNormDiffL2GetBufferHostSize_16u_C4R . . . . .   | 1727 |
| 7.103.2.35 nppiNormDiffL2GetBufferHostSize_32f_AC4R . . . . .  | 1727 |
| 7.103.2.36 nppiNormDiffL2GetBufferHostSize_32f_C1MR . . . . .  | 1727 |
| 7.103.2.37 nppiNormDiffL2GetBufferHostSize_32f_C1R . . . . .   | 1727 |
| 7.103.2.38 nppiNormDiffL2GetBufferHostSize_32f_C3CMR . . . . . | 1728 |
| 7.103.2.39 nppiNormDiffL2GetBufferHostSize_32f_C3R . . . . .   | 1728 |
| 7.103.2.40 nppiNormDiffL2GetBufferHostSize_32f_C4R . . . . .   | 1728 |
| 7.103.2.41 nppiNormDiffL2GetBufferHostSize_8s_C1MR . . . . .   | 1729 |
| 7.103.2.42 nppiNormDiffL2GetBufferHostSize_8s_C3CMR . . . . .  | 1729 |

|            |  |      |
|------------|--|------|
| 7.103.2.43 | nppiNormDiffL2GetBufferHostSize_8u_AC4R  | 1729 |
| 7.103.2.44 | nppiNormDiffL2GetBufferHostSize_8u_C1MR  | 1729 |
| 7.103.2.45 | nppiNormDiffL2GetBufferHostSize_8u_C1R   | 1730 |
| 7.103.2.46 | nppiNormDiffL2GetBufferHostSize_8u_C3CMR | 1730 |
| 7.103.2.47 | nppiNormDiffL2GetBufferHostSize_8u_C3R   | 1730 |
| 7.103.2.48 | nppiNormDiffL2GetBufferHostSize_8u_C4R   | 1731 |
| 7.104      | NormRel_Inf                              | 1732 |
| 7.104.1    | Detailed Description                     | 1736 |
| 7.104.2    | Function Documentation                   | 1736 |
| 7.104.2.1  | nppiNormRel_Inf_16s_AC4R                 | 1736 |
| 7.104.2.2  | nppiNormRel_Inf_16s_C1R                  | 1736 |
| 7.104.2.3  | nppiNormRel_Inf_16s_C3R                  | 1737 |
| 7.104.2.4  | nppiNormRel_Inf_16s_C4R                  | 1737 |
| 7.104.2.5  | nppiNormRel_Inf_16u_AC4R                 | 1738 |
| 7.104.2.6  | nppiNormRel_Inf_16u_C1MR                 | 1738 |
| 7.104.2.7  | nppiNormRel_Inf_16u_C1R                  | 1739 |
| 7.104.2.8  | nppiNormRel_Inf_16u_C3CMR                | 1739 |
| 7.104.2.9  | nppiNormRel_Inf_16u_C3R                  | 1740 |
| 7.104.2.10 | nppiNormRel_Inf_16u_C4R                  | 1740 |
| 7.104.2.11 | nppiNormRel_Inf_32f_AC4R                 | 1741 |
| 7.104.2.12 | nppiNormRel_Inf_32f_C1MR                 | 1741 |
| 7.104.2.13 | nppiNormRel_Inf_32f_C1R                  | 1742 |
| 7.104.2.14 | nppiNormRel_Inf_32f_C3CMR                | 1742 |
| 7.104.2.15 | nppiNormRel_Inf_32f_C3R                  | 1743 |
| 7.104.2.16 | nppiNormRel_Inf_32f_C4R                  | 1743 |
| 7.104.2.17 | nppiNormRel_Inf_8s_C1MR                  | 1744 |
| 7.104.2.18 | nppiNormRel_Inf_8s_C3CMR                 | 1744 |
| 7.104.2.19 | nppiNormRel_Inf_8u_AC4R                  | 1745 |
| 7.104.2.20 | nppiNormRel_Inf_8u_C1MR                  | 1745 |
| 7.104.2.21 | nppiNormRel_Inf_8u_C1R                   | 1746 |
| 7.104.2.22 | nppiNormRel_Inf_8u_C3CMR                 | 1746 |
| 7.104.2.23 | nppiNormRel_Inf_8u_C3R                   | 1747 |
| 7.104.2.24 | nppiNormRel_Inf_8u_C4R                   | 1747 |
| 7.104.2.25 | nppiNormRelInfGetBufferHostSize_16s_AC4R | 1747 |
| 7.104.2.26 | nppiNormRelInfGetBufferHostSize_16s_C1R  | 1748 |
| 7.104.2.27 | nppiNormRelInfGetBufferHostSize_16s_C3R  | 1748 |

|            |   |      |
|------------|---|------|
| 7.104.2.28 | nppiNormRelInfGetBufferHostSize_16s_C4R . . . . .   | 1748 |
| 7.104.2.29 | nppiNormRelInfGetBufferHostSize_16u_AC4R . . . . .  | 1749 |
| 7.104.2.30 | nppiNormRelInfGetBufferHostSize_16u_C1MR . . . . .  | 1749 |
| 7.104.2.31 | nppiNormRelInfGetBufferHostSize_16u_C1R . . . . .   | 1749 |
| 7.104.2.32 | nppiNormRelInfGetBufferHostSize_16u_C3CMR . . . . . | 1749 |
| 7.104.2.33 | nppiNormRelInfGetBufferHostSize_16u_C3R . . . . .   | 1750 |
| 7.104.2.34 | nppiNormRelInfGetBufferHostSize_16u_C4R . . . . .   | 1750 |
| 7.104.2.35 | nppiNormRelInfGetBufferHostSize_32f_AC4R . . . . .  | 1750 |
| 7.104.2.36 | nppiNormRelInfGetBufferHostSize_32f_C1MR . . . . .  | 1751 |
| 7.104.2.37 | nppiNormRelInfGetBufferHostSize_32f_C1R . . . . .   | 1751 |
| 7.104.2.38 | nppiNormRelInfGetBufferHostSize_32f_C3CMR . . . . . | 1751 |
| 7.104.2.39 | nppiNormRelInfGetBufferHostSize_32f_C3R . . . . .   | 1751 |
| 7.104.2.40 | nppiNormRelInfGetBufferHostSize_32f_C4R . . . . .   | 1752 |
| 7.104.2.41 | nppiNormRelInfGetBufferHostSize_32s_C1R . . . . .   | 1752 |
| 7.104.2.42 | nppiNormRelInfGetBufferHostSize_8s_C1MR . . . . .   | 1752 |
| 7.104.2.43 | nppiNormRelInfGetBufferHostSize_8s_C3CMR . . . . .  | 1753 |
| 7.104.2.44 | nppiNormRelInfGetBufferHostSize_8u_AC4R . . . . .   | 1753 |
| 7.104.2.45 | nppiNormRelInfGetBufferHostSize_8u_C1MR . . . . .   | 1753 |
| 7.104.2.46 | nppiNormRelInfGetBufferHostSize_8u_C1R . . . . .    | 1753 |
| 7.104.2.47 | nppiNormRelInfGetBufferHostSize_8u_C3CMR . . . . .  | 1754 |
| 7.104.2.48 | nppiNormRelInfGetBufferHostSize_8u_C3R . . . . .    | 1754 |
| 7.104.2.49 | nppiNormRelInfGetBufferHostSize_8u_C4R . . . . .    | 1754 |
| 7.105      | NormRel_L1 . . . . .                                | 1755 |
| 7.105.1    | Detailed Description . . . . .                      | 1759 |
| 7.105.2    | Function Documentation . . . . .                    | 1759 |
| 7.105.2.1  | nppiNormRel_L1_16s_AC4R . . . . .                   | 1759 |
| 7.105.2.2  | nppiNormRel_L1_16s_C1R . . . . .                    | 1759 |
| 7.105.2.3  | nppiNormRel_L1_16s_C3R . . . . .                    | 1760 |
| 7.105.2.4  | nppiNormRel_L1_16s_C4R . . . . .                    | 1760 |
| 7.105.2.5  | nppiNormRel_L1_16u_AC4R . . . . .                   | 1761 |
| 7.105.2.6  | nppiNormRel_L1_16u_C1MR . . . . .                   | 1761 |
| 7.105.2.7  | nppiNormRel_L1_16u_C1R . . . . .                    | 1762 |
| 7.105.2.8  | nppiNormRel_L1_16u_C3CMR . . . . .                  | 1762 |
| 7.105.2.9  | nppiNormRel_L1_16u_C3R . . . . .                    | 1763 |
| 7.105.2.10 | nppiNormRel_L1_16u_C4R . . . . .                    | 1763 |
| 7.105.2.11 | nppiNormRel_L1_32f_AC4R . . . . .                   | 1763 |



|  |      |
|--|------|
| 7.105.2.12nppiNormRel_L1_32f_C1MR . . . . .                  | 1764 |
| 7.105.2.13nppiNormRel_L1_32f_C1R . . . . .                   | 1764 |
| 7.105.2.14nppiNormRel_L1_32f_C3CMR . . . . .                 | 1765 |
| 7.105.2.15nppiNormRel_L1_32f_C3R . . . . .                   | 1765 |
| 7.105.2.16nppiNormRel_L1_32f_C4R . . . . .                   | 1766 |
| 7.105.2.17nppiNormRel_L1_8s_C1MR . . . . .                   | 1766 |
| 7.105.2.18nppiNormRel_L1_8s_C3CMR . . . . .                  | 1767 |
| 7.105.2.19nppiNormRel_L1_8u_AC4R . . . . .                   | 1767 |
| 7.105.2.20nppiNormRel_L1_8u_C1MR . . . . .                   | 1768 |
| 7.105.2.21nppiNormRel_L1_8u_C1R . . . . .                    | 1768 |
| 7.105.2.22nppiNormRel_L1_8u_C3CMR . . . . .                  | 1769 |
| 7.105.2.23nppiNormRel_L1_8u_C3R . . . . .                    | 1769 |
| 7.105.2.24nppiNormRel_L1_8u_C4R . . . . .                    | 1770 |
| 7.105.2.25nppiNormRelL1GetBufferHostSize_16s_AC4R . . . . .  | 1770 |
| 7.105.2.26nppiNormRelL1GetBufferHostSize_16s_C1R . . . . .   | 1771 |
| 7.105.2.27nppiNormRelL1GetBufferHostSize_16s_C3R . . . . .   | 1771 |
| 7.105.2.28nppiNormRelL1GetBufferHostSize_16s_C4R . . . . .   | 1771 |
| 7.105.2.29nppiNormRelL1GetBufferHostSize_16u_AC4R . . . . .  | 1771 |
| 7.105.2.30nppiNormRelL1GetBufferHostSize_16u_C1MR . . . . .  | 1772 |
| 7.105.2.31nppiNormRelL1GetBufferHostSize_16u_C1R . . . . .   | 1772 |
| 7.105.2.32nppiNormRelL1GetBufferHostSize_16u_C3CMR . . . . . | 1772 |
| 7.105.2.33nppiNormRelL1GetBufferHostSize_16u_C3R . . . . .   | 1773 |
| 7.105.2.34nppiNormRelL1GetBufferHostSize_16u_C4R . . . . .   | 1773 |
| 7.105.2.35nppiNormRelL1GetBufferHostSize_32f_AC4R . . . . .  | 1773 |
| 7.105.2.36nppiNormRelL1GetBufferHostSize_32f_C1MR . . . . .  | 1773 |
| 7.105.2.37nppiNormRelL1GetBufferHostSize_32f_C1R . . . . .   | 1774 |
| 7.105.2.38nppiNormRelL1GetBufferHostSize_32f_C3CMR . . . . . | 1774 |
| 7.105.2.39nppiNormRelL1GetBufferHostSize_32f_C3R . . . . .   | 1774 |
| 7.105.2.40nppiNormRelL1GetBufferHostSize_32f_C4R . . . . .   | 1775 |
| 7.105.2.41nppiNormRelL1GetBufferHostSize_8s_C1MR . . . . .   | 1775 |
| 7.105.2.42nppiNormRelL1GetBufferHostSize_8s_C3CMR . . . . .  | 1775 |
| 7.105.2.43nppiNormRelL1GetBufferHostSize_8u_AC4R . . . . .   | 1775 |
| 7.105.2.44nppiNormRelL1GetBufferHostSize_8u_C1MR . . . . .   | 1776 |
| 7.105.2.45nppiNormRelL1GetBufferHostSize_8u_C1R . . . . .    | 1776 |
| 7.105.2.46nppiNormRelL1GetBufferHostSize_8u_C3CMR . . . . .  | 1776 |
| 7.105.2.47nppiNormRelL1GetBufferHostSize_8u_C3R . . . . .    | 1777 |

|            |  |      |
|------------|--|------|
| 7.105.2.4  | <a href="#">nppiNormRelL1GetBufferHostSize_8u_C4R</a>    | 1777 |
| 7.106      | <a href="#">NormRel_L2</a>                               | 1778 |
| 7.106.1    | <a href="#">Detailed Description</a>                     | 1782 |
| 7.106.2    | <a href="#">Function Documentation</a>                   | 1782 |
| 7.106.2.1  | <a href="#">nppiNormRel_L2_16s_AC4R</a>                  | 1782 |
| 7.106.2.2  | <a href="#">nppiNormRel_L2_16s_C1R</a>                   | 1782 |
| 7.106.2.3  | <a href="#">nppiNormRel_L2_16s_C3R</a>                   | 1783 |
| 7.106.2.4  | <a href="#">nppiNormRel_L2_16s_C4R</a>                   | 1783 |
| 7.106.2.5  | <a href="#">nppiNormRel_L2_16u_AC4R</a>                  | 1784 |
| 7.106.2.6  | <a href="#">nppiNormRel_L2_16u_C1MR</a>                  | 1784 |
| 7.106.2.7  | <a href="#">nppiNormRel_L2_16u_C1R</a>                   | 1785 |
| 7.106.2.8  | <a href="#">nppiNormRel_L2_16u_C3CMR</a>                 | 1785 |
| 7.106.2.9  | <a href="#">nppiNormRel_L2_16u_C3R</a>                   | 1786 |
| 7.106.2.10 | <a href="#">nppiNormRel_L2_16u_C4R</a>                   | 1786 |
| 7.106.2.11 | <a href="#">nppiNormRel_L2_32f_AC4R</a>                  | 1786 |
| 7.106.2.12 | <a href="#">nppiNormRel_L2_32f_C1MR</a>                  | 1787 |
| 7.106.2.13 | <a href="#">nppiNormRel_L2_32f_C1R</a>                   | 1787 |
| 7.106.2.14 | <a href="#">nppiNormRel_L2_32f_C3CMR</a>                 | 1788 |
| 7.106.2.15 | <a href="#">nppiNormRel_L2_32f_C3R</a>                   | 1788 |
| 7.106.2.16 | <a href="#">nppiNormRel_L2_32f_C4R</a>                   | 1789 |
| 7.106.2.17 | <a href="#">nppiNormRel_L2_8s_C1MR</a>                   | 1789 |
| 7.106.2.18 | <a href="#">nppiNormRel_L2_8s_C3CMR</a>                  | 1790 |
| 7.106.2.19 | <a href="#">nppiNormRel_L2_8u_AC4R</a>                   | 1790 |
| 7.106.2.20 | <a href="#">nppiNormRel_L2_8u_C1MR</a>                   | 1791 |
| 7.106.2.21 | <a href="#">nppiNormRel_L2_8u_C1R</a>                    | 1791 |
| 7.106.2.22 | <a href="#">nppiNormRel_L2_8u_C3CMR</a>                  | 1792 |
| 7.106.2.23 | <a href="#">nppiNormRel_L2_8u_C3R</a>                    | 1792 |
| 7.106.2.24 | <a href="#">nppiNormRel_L2_8u_C4R</a>                    | 1793 |
| 7.106.2.25 | <a href="#">nppiNormRelL2GetBufferHostSize_16s_AC4R</a>  | 1793 |
| 7.106.2.26 | <a href="#">nppiNormRelL2GetBufferHostSize_16s_C1R</a>   | 1794 |
| 7.106.2.27 | <a href="#">nppiNormRelL2GetBufferHostSize_16s_C3R</a>   | 1794 |
| 7.106.2.28 | <a href="#">nppiNormRelL2GetBufferHostSize_16s_C4R</a>   | 1794 |
| 7.106.2.29 | <a href="#">nppiNormRelL2GetBufferHostSize_16u_AC4R</a>  | 1794 |
| 7.106.2.30 | <a href="#">nppiNormRelL2GetBufferHostSize_16u_C1MR</a>  | 1795 |
| 7.106.2.31 | <a href="#">nppiNormRelL2GetBufferHostSize_16u_C1R</a>   | 1795 |
| 7.106.2.32 | <a href="#">nppiNormRelL2GetBufferHostSize_16u_C3CMR</a> | 1795 |

|            |  |      |
|------------|--|------|
| 7.106.2.33 | nppiNormRelL2GetBufferHostSize_16u_C3R . . . . .   | 1796 |
| 7.106.2.34 | nppiNormRelL2GetBufferHostSize_16u_C4R . . . . .   | 1796 |
| 7.106.2.35 | nppiNormRelL2GetBufferHostSize_32f_AC4R . . . . .  | 1796 |
| 7.106.2.36 | nppiNormRelL2GetBufferHostSize_32f_C1MR . . . . .  | 1796 |
| 7.106.2.37 | nppiNormRelL2GetBufferHostSize_32f_C1R . . . . .   | 1797 |
| 7.106.2.38 | nppiNormRelL2GetBufferHostSize_32f_C3CMR . . . . . | 1797 |
| 7.106.2.39 | nppiNormRelL2GetBufferHostSize_32f_C3R . . . . .   | 1797 |
| 7.106.2.40 | nppiNormRelL2GetBufferHostSize_32f_C4R . . . . .   | 1798 |
| 7.106.2.41 | nppiNormRelL2GetBufferHostSize_8s_C1MR . . . . .   | 1798 |
| 7.106.2.42 | nppiNormRelL2GetBufferHostSize_8s_C3CMR . . . . .  | 1798 |
| 7.106.2.43 | nppiNormRelL2GetBufferHostSize_8u_AC4R . . . . .   | 1798 |
| 7.106.2.44 | nppiNormRelL2GetBufferHostSize_8u_C1MR . . . . .   | 1799 |
| 7.106.2.45 | nppiNormRelL2GetBufferHostSize_8u_C1R . . . . .    | 1799 |
| 7.106.2.46 | nppiNormRelL2GetBufferHostSize_8u_C3CMR . . . . .  | 1799 |
| 7.106.2.47 | nppiNormRelL2GetBufferHostSize_8u_C3R . . . . .    | 1800 |
| 7.106.2.48 | nppiNormRelL2GetBufferHostSize_8u_C4R . . . . .    | 1800 |
| 7.107      | DotProd . . . . .                                  | 1801 |
| 7.107.1    | Detailed Description . . . . .                     | 1805 |
| 7.107.2    | Function Documentation . . . . .                   | 1805 |
| 7.107.2.1  | nppiDotProd_16s64f_AC4R . . . . .                  | 1805 |
| 7.107.2.2  | nppiDotProd_16s64f_C1R . . . . .                   | 1806 |
| 7.107.2.3  | nppiDotProd_16s64f_C3R . . . . .                   | 1806 |
| 7.107.2.4  | nppiDotProd_16s64f_C4R . . . . .                   | 1806 |
| 7.107.2.5  | nppiDotProd_16u64f_AC4R . . . . .                  | 1807 |
| 7.107.2.6  | nppiDotProd_16u64f_C1R . . . . .                   | 1807 |
| 7.107.2.7  | nppiDotProd_16u64f_C3R . . . . .                   | 1808 |
| 7.107.2.8  | nppiDotProd_16u64f_C4R . . . . .                   | 1808 |
| 7.107.2.9  | nppiDotProd_32f64f_AC4R . . . . .                  | 1809 |
| 7.107.2.10 | nppiDotProd_32f64f_C1R . . . . .                   | 1809 |
| 7.107.2.11 | nppiDotProd_32f64f_C3R . . . . .                   | 1809 |
| 7.107.2.12 | nppiDotProd_32f64f_C4R . . . . .                   | 1810 |
| 7.107.2.13 | nppiDotProd_32s64f_AC4R . . . . .                  | 1810 |
| 7.107.2.14 | nppiDotProd_32s64f_C1R . . . . .                   | 1811 |
| 7.107.2.15 | nppiDotProd_32s64f_C3R . . . . .                   | 1811 |
| 7.107.2.16 | nppiDotProd_32s64f_C4R . . . . .                   | 1812 |
| 7.107.2.17 | nppiDotProd_32u64f_AC4R . . . . .                  | 1812 |

|  |      |
|--|------|
| 7.107.2.18nppiDotProd_32u64f_C1R . . . . .                   | 1812 |
| 7.107.2.19nppiDotProd_32u64f_C3R . . . . .                   | 1813 |
| 7.107.2.20nppiDotProd_32u64f_C4R . . . . .                   | 1813 |
| 7.107.2.21nppiDotProd_8s64f_AC4R . . . . .                   | 1814 |
| 7.107.2.22nppiDotProd_8s64f_C1R . . . . .                    | 1814 |
| 7.107.2.23nppiDotProd_8s64f_C3R . . . . .                    | 1815 |
| 7.107.2.24nppiDotProd_8s64f_C4R . . . . .                    | 1815 |
| 7.107.2.25nppiDotProd_8u64f_AC4R . . . . .                   | 1815 |
| 7.107.2.26nppiDotProd_8u64f_C1R . . . . .                    | 1816 |
| 7.107.2.27nppiDotProd_8u64f_C3R . . . . .                    | 1816 |
| 7.107.2.28nppiDotProd_8u64f_C4R . . . . .                    | 1817 |
| 7.107.2.29nppiDotProdGetBufferHostSize_16s64f_AC4R . . . . . | 1817 |
| 7.107.2.30nppiDotProdGetBufferHostSize_16s64f_C1R . . . . .  | 1817 |
| 7.107.2.31nppiDotProdGetBufferHostSize_16s64f_C3R . . . . .  | 1818 |
| 7.107.2.32nppiDotProdGetBufferHostSize_16s64f_C4R . . . . .  | 1818 |
| 7.107.2.33nppiDotProdGetBufferHostSize_16u64f_AC4R . . . . . | 1818 |
| 7.107.2.34nppiDotProdGetBufferHostSize_16u64f_C1R . . . . .  | 1819 |
| 7.107.2.35nppiDotProdGetBufferHostSize_16u64f_C3R . . . . .  | 1819 |
| 7.107.2.36nppiDotProdGetBufferHostSize_16u64f_C4R . . . . .  | 1819 |
| 7.107.2.37nppiDotProdGetBufferHostSize_32f64f_AC4R . . . . . | 1819 |
| 7.107.2.38nppiDotProdGetBufferHostSize_32f64f_C1R . . . . .  | 1820 |
| 7.107.2.39nppiDotProdGetBufferHostSize_32f64f_C3R . . . . .  | 1820 |
| 7.107.2.40nppiDotProdGetBufferHostSize_32f64f_C4R . . . . .  | 1820 |
| 7.107.2.41nppiDotProdGetBufferHostSize_32s64f_AC4R . . . . . | 1821 |
| 7.107.2.42nppiDotProdGetBufferHostSize_32s64f_C1R . . . . .  | 1821 |
| 7.107.2.43nppiDotProdGetBufferHostSize_32s64f_C3R . . . . .  | 1821 |
| 7.107.2.44nppiDotProdGetBufferHostSize_32s64f_C4R . . . . .  | 1821 |
| 7.107.2.45nppiDotProdGetBufferHostSize_32u64f_AC4R . . . . . | 1822 |
| 7.107.2.46nppiDotProdGetBufferHostSize_32u64f_C1R . . . . .  | 1822 |
| 7.107.2.47nppiDotProdGetBufferHostSize_32u64f_C3R . . . . .  | 1822 |
| 7.107.2.48nppiDotProdGetBufferHostSize_32u64f_C4R . . . . .  | 1823 |
| 7.107.2.49nppiDotProdGetBufferHostSize_8s64f_AC4R . . . . .  | 1823 |
| 7.107.2.50nppiDotProdGetBufferHostSize_8s64f_C1R . . . . .   | 1823 |
| 7.107.2.51nppiDotProdGetBufferHostSize_8s64f_C3R . . . . .   | 1823 |
| 7.107.2.52nppiDotProdGetBufferHostSize_8s64f_C4R . . . . .   | 1824 |
| 7.107.2.53nppiDotProdGetBufferHostSize_8u64f_AC4R . . . . .  | 1824 |

|            |  |      |
|------------|--|------|
| 7.107.2.54 | <a href="#">nppiDotProdGetBufferHostSize_8u64f_C1R</a>     | 1824 |
| 7.107.2.55 | <a href="#">nppiDotProdGetBufferHostSize_8u64f_C3R</a>     | 1825 |
| 7.107.2.56 | <a href="#">nppiDotProdGetBufferHostSize_8u64f_C4R</a>     | 1825 |
| 7.108      | <a href="#">CountInRange</a>                               | 1826 |
| 7.108.1    | Detailed Description                                       | 1827 |
| 7.108.2    | Function Documentation                                     | 1827 |
| 7.108.2.1  | <a href="#">nppiCountInRange_32f_AC4R</a>                  | 1827 |
| 7.108.2.2  | <a href="#">nppiCountInRange_32f_C1R</a>                   | 1827 |
| 7.108.2.3  | <a href="#">nppiCountInRange_32f_C3R</a>                   | 1828 |
| 7.108.2.4  | <a href="#">nppiCountInRange_8u_AC4R</a>                   | 1828 |
| 7.108.2.5  | <a href="#">nppiCountInRange_8u_C1R</a>                    | 1829 |
| 7.108.2.6  | <a href="#">nppiCountInRange_8u_C3R</a>                    | 1829 |
| 7.108.2.7  | <a href="#">nppiCountInRangeGetBufferHostSize_32f_AC4R</a> | 1830 |
| 7.108.2.8  | <a href="#">nppiCountInRangeGetBufferHostSize_32f_C1R</a>  | 1830 |
| 7.108.2.9  | <a href="#">nppiCountInRangeGetBufferHostSize_32f_C3R</a>  | 1830 |
| 7.108.2.10 | <a href="#">nppiCountInRangeGetBufferHostSize_8u_AC4R</a>  | 1831 |
| 7.108.2.11 | <a href="#">nppiCountInRangeGetBufferHostSize_8u_C1R</a>   | 1831 |
| 7.108.2.12 | <a href="#">nppiCountInRangeGetBufferHostSize_8u_C3R</a>   | 1831 |
| 7.109      | <a href="#">MaxEvery</a>                                   | 1832 |
| 7.109.1    | Detailed Description                                       | 1833 |
| 7.109.2    | Function Documentation                                     | 1833 |
| 7.109.2.1  | <a href="#">nppiMaxEvery_16s_AC4IR</a>                     | 1833 |
| 7.109.2.2  | <a href="#">nppiMaxEvery_16s_C1IR</a>                      | 1834 |
| 7.109.2.3  | <a href="#">nppiMaxEvery_16s_C3IR</a>                      | 1834 |
| 7.109.2.4  | <a href="#">nppiMaxEvery_16s_C4IR</a>                      | 1834 |
| 7.109.2.5  | <a href="#">nppiMaxEvery_16u_AC4IR</a>                     | 1835 |
| 7.109.2.6  | <a href="#">nppiMaxEvery_16u_C1IR</a>                      | 1835 |
| 7.109.2.7  | <a href="#">nppiMaxEvery_16u_C3IR</a>                      | 1835 |
| 7.109.2.8  | <a href="#">nppiMaxEvery_16u_C4IR</a>                      | 1836 |
| 7.109.2.9  | <a href="#">nppiMaxEvery_32f_AC4IR</a>                     | 1836 |
| 7.109.2.10 | <a href="#">nppiMaxEvery_32f_C1IR</a>                      | 1836 |
| 7.109.2.11 | <a href="#">nppiMaxEvery_32f_C3IR</a>                      | 1837 |
| 7.109.2.12 | <a href="#">nppiMaxEvery_32f_C4IR</a>                      | 1837 |
| 7.109.2.13 | <a href="#">nppiMaxEvery_8u_AC4IR</a>                      | 1837 |
| 7.109.2.14 | <a href="#">nppiMaxEvery_8u_C1IR</a>                       | 1838 |
| 7.109.2.15 | <a href="#">nppiMaxEvery_8u_C3IR</a>                       | 1838 |

|  |      |
|--|------|
| 7.109.2.16 nppiMaxEvery_8u_C4IR        | 1838 |
| 7.110MinEvery                          | 1839 |
| 7.110.1 Detailed Description           | 1840 |
| 7.110.2 Function Documentation         | 1840 |
| 7.110.2.1 nppiMinEvery_16s_AC4IR       | 1840 |
| 7.110.2.2 nppiMinEvery_16s_C1IR        | 1841 |
| 7.110.2.3 nppiMinEvery_16s_C3IR        | 1841 |
| 7.110.2.4 nppiMinEvery_16s_C4IR        | 1841 |
| 7.110.2.5 nppiMinEvery_16u_AC4IR       | 1842 |
| 7.110.2.6 nppiMinEvery_16u_C1IR        | 1842 |
| 7.110.2.7 nppiMinEvery_16u_C3IR        | 1842 |
| 7.110.2.8 nppiMinEvery_16u_C4IR        | 1843 |
| 7.110.2.9 nppiMinEvery_32f_AC4IR       | 1843 |
| 7.110.2.10 nppiMinEvery_32f_C1IR       | 1843 |
| 7.110.2.11 nppiMinEvery_32f_C3IR       | 1844 |
| 7.110.2.12 nppiMinEvery_32f_C4IR       | 1844 |
| 7.110.2.13 nppiMinEvery_8u_AC4IR       | 1844 |
| 7.110.2.14 nppiMinEvery_8u_C1IR        | 1845 |
| 7.110.2.15 nppiMinEvery_8u_C3IR        | 1845 |
| 7.110.2.16 nppiMinEvery_8u_C4IR        | 1845 |
| 7.111Integral                          | 1846 |
| 7.111.1 Detailed Description           | 1846 |
| 7.111.2 Function Documentation         | 1846 |
| 7.111.2.1 nppiIntegral_8u32f_C1R       | 1846 |
| 7.111.2.2 nppiIntegral_8u32s_C1R       | 1847 |
| 7.112SqrIntegral                       | 1848 |
| 7.112.1 Detailed Description           | 1848 |
| 7.112.2 Function Documentation         | 1848 |
| 7.112.2.1 nppiSqrIntegral_8u32f64f_C1R | 1848 |
| 7.112.2.2 nppiSqrIntegral_8u32s64f_C1R | 1849 |
| 7.112.2.3 nppiSqrIntegral_8u32s_C1R    | 1849 |
| 7.113RectStdDev                        | 1851 |
| 7.113.1 Detailed Description           | 1851 |
| 7.113.2 Function Documentation         | 1851 |
| 7.113.2.1 nppiRectStdDev_32f_C1R       | 1851 |
| 7.113.2.2 nppiRectStdDev_32s32f_C1R    | 1852 |

|   |      |
|---|------|
| 7.113.2.3 nppiRectStdDev_32s_C1RSfs . . . . .               | 1852 |
| 7.114HistogramEven . . . . .                                | 1854 |
| 7.114.1 Detailed Description . . . . .                      | 1856 |
| 7.114.2 Function Documentation . . . . .                    | 1856 |
| 7.114.2.1 nppiEvenLevelsHost_32s . . . . .                  | 1856 |
| 7.114.2.2 nppiHistogramEven_16s_AC4R . . . . .              | 1857 |
| 7.114.2.3 nppiHistogramEven_16s_C1R . . . . .               | 1857 |
| 7.114.2.4 nppiHistogramEven_16s_C3R . . . . .               | 1858 |
| 7.114.2.5 nppiHistogramEven_16s_C4R . . . . .               | 1858 |
| 7.114.2.6 nppiHistogramEven_16u_AC4R . . . . .              | 1859 |
| 7.114.2.7 nppiHistogramEven_16u_C1R . . . . .               | 1859 |
| 7.114.2.8 nppiHistogramEven_16u_C3R . . . . .               | 1860 |
| 7.114.2.9 nppiHistogramEven_16u_C4R . . . . .               | 1860 |
| 7.114.2.10nppiHistogramEven_8u_AC4R . . . . .               | 1861 |
| 7.114.2.11nppiHistogramEven_8u_C1R . . . . .                | 1861 |
| 7.114.2.12nppiHistogramEven_8u_C3R . . . . .                | 1861 |
| 7.114.2.13nppiHistogramEven_8u_C4R . . . . .                | 1862 |
| 7.114.2.14nppiHistogramEvenGetBufferSize_16s_AC4R . . . . . | 1862 |
| 7.114.2.15nppiHistogramEvenGetBufferSize_16s_C1R . . . . .  | 1863 |
| 7.114.2.16nppiHistogramEvenGetBufferSize_16s_C3R . . . . .  | 1863 |
| 7.114.2.17nppiHistogramEvenGetBufferSize_16s_C4R . . . . .  | 1863 |
| 7.114.2.18nppiHistogramEvenGetBufferSize_16u_AC4R . . . . . | 1864 |
| 7.114.2.19nppiHistogramEvenGetBufferSize_16u_C1R . . . . .  | 1864 |
| 7.114.2.20nppiHistogramEvenGetBufferSize_16u_C3R . . . . .  | 1864 |
| 7.114.2.21nppiHistogramEvenGetBufferSize_16u_C4R . . . . .  | 1865 |
| 7.114.2.22nppiHistogramEvenGetBufferSize_8u_AC4R . . . . .  | 1865 |
| 7.114.2.23nppiHistogramEvenGetBufferSize_8u_C1R . . . . .   | 1865 |
| 7.114.2.24nppiHistogramEvenGetBufferSize_8u_C3R . . . . .   | 1866 |
| 7.114.2.25nppiHistogramEvenGetBufferSize_8u_C4R . . . . .   | 1866 |
| 7.115HistogramRange . . . . .                               | 1867 |
| 7.115.1 Detailed Description . . . . .                      | 1869 |
| 7.115.2 Function Documentation . . . . .                    | 1870 |
| 7.115.2.1 nppiHistogramRange_16s_AC4R . . . . .             | 1870 |
| 7.115.2.2 nppiHistogramRange_16s_C1R . . . . .              | 1870 |
| 7.115.2.3 nppiHistogramRange_16s_C3R . . . . .              | 1871 |
| 7.115.2.4 nppiHistogramRange_16s_C4R . . . . .              | 1871 |

|   |      |
|---|------|
| 7.115.2.5 nppiHistogramRange_16u_AC4R . . . . .               | 1871 |
| 7.115.2.6 nppiHistogramRange_16u_C1R . . . . .                | 1872 |
| 7.115.2.7 nppiHistogramRange_16u_C3R . . . . .                | 1872 |
| 7.115.2.8 nppiHistogramRange_16u_C4R . . . . .                | 1873 |
| 7.115.2.9 nppiHistogramRange_32f_AC4R . . . . .               | 1873 |
| 7.115.2.10 nppiHistogramRange_32f_C1R . . . . .               | 1874 |
| 7.115.2.11 nppiHistogramRange_32f_C3R . . . . .               | 1874 |
| 7.115.2.12 nppiHistogramRange_32f_C4R . . . . .               | 1875 |
| 7.115.2.13 nppiHistogramRange_8u_AC4R . . . . .               | 1875 |
| 7.115.2.14 nppiHistogramRange_8u_C1R . . . . .                | 1876 |
| 7.115.2.15 nppiHistogramRange_8u_C3R . . . . .                | 1876 |
| 7.115.2.16 nppiHistogramRange_8u_C4R . . . . .                | 1876 |
| 7.115.2.17 nppiHistogramRangeGetBufferSize_16s_AC4R . . . . . | 1877 |
| 7.115.2.18 nppiHistogramRangeGetBufferSize_16s_C1R . . . . .  | 1877 |
| 7.115.2.19 nppiHistogramRangeGetBufferSize_16s_C3R . . . . .  | 1878 |
| 7.115.2.20 nppiHistogramRangeGetBufferSize_16s_C4R . . . . .  | 1878 |
| 7.115.2.21 nppiHistogramRangeGetBufferSize_16u_AC4R . . . . . | 1878 |
| 7.115.2.22 nppiHistogramRangeGetBufferSize_16u_C1R . . . . .  | 1879 |
| 7.115.2.23 nppiHistogramRangeGetBufferSize_16u_C3R . . . . .  | 1879 |
| 7.115.2.24 nppiHistogramRangeGetBufferSize_16u_C4R . . . . .  | 1879 |
| 7.115.2.25 nppiHistogramRangeGetBufferSize_32f_AC4R . . . . . | 1880 |
| 7.115.2.26 nppiHistogramRangeGetBufferSize_32f_C1R . . . . .  | 1880 |
| 7.115.2.27 nppiHistogramRangeGetBufferSize_32f_C3R . . . . .  | 1880 |
| 7.115.2.28 nppiHistogramRangeGetBufferSize_32f_C4R . . . . .  | 1881 |
| 7.115.2.29 nppiHistogramRangeGetBufferSize_8u_AC4R . . . . .  | 1881 |
| 7.115.2.30 nppiHistogramRangeGetBufferSize_8u_C1R . . . . .   | 1881 |
| 7.115.2.31 nppiHistogramRangeGetBufferSize_8u_C3R . . . . .   | 1882 |
| 7.115.2.32 nppiHistogramRangeGetBufferSize_8u_C4R . . . . .   | 1882 |
| 7.116 Image Proximity . . . . .                               | 1883 |
| 7.116.1 Detailed Description . . . . .                        | 1883 |
| 7.116.2 General Introduction . . . . .                        | 1883 |
| 7.116.3 Categorizations . . . . .                             | 1885 |
| 7.117 SqrDistanceFull_Norm . . . . .                          | 1886 |
| 7.117.1 Detailed Description . . . . .                        | 1887 |
| 7.117.2 Function Documentation . . . . .                      | 1888 |
| 7.117.2.1 nppiSqrDistanceFull_Norm_16u32f_AC4R . . . . .      | 1888 |



|            |  |      |
|------------|--|------|
| 7.117.2.2  | <a href="#">nppiSqrDistanceFull_Norm_16u32f_C1R</a>  | 1888 |
| 7.117.2.3  | <a href="#">nppiSqrDistanceFull_Norm_16u32f_C3R</a>  | 1889 |
| 7.117.2.4  | <a href="#">nppiSqrDistanceFull_Norm_16u32f_C4R</a>  | 1889 |
| 7.117.2.5  | <a href="#">nppiSqrDistanceFull_Norm_32f_AC4R</a>    | 1889 |
| 7.117.2.6  | <a href="#">nppiSqrDistanceFull_Norm_32f_C1R</a>     | 1890 |
| 7.117.2.7  | <a href="#">nppiSqrDistanceFull_Norm_32f_C3R</a>     | 1890 |
| 7.117.2.8  | <a href="#">nppiSqrDistanceFull_Norm_32f_C4R</a>     | 1891 |
| 7.117.2.9  | <a href="#">nppiSqrDistanceFull_Norm_8s32f_AC4R</a>  | 1891 |
| 7.117.2.10 | <a href="#">nppiSqrDistanceFull_Norm_8s32f_C1R</a>   | 1892 |
| 7.117.2.11 | <a href="#">nppiSqrDistanceFull_Norm_8s32f_C3R</a>   | 1892 |
| 7.117.2.12 | <a href="#">nppiSqrDistanceFull_Norm_8s32f_C4R</a>   | 1892 |
| 7.117.2.13 | <a href="#">nppiSqrDistanceFull_Norm_8u32f_AC4R</a>  | 1893 |
| 7.117.2.14 | <a href="#">nppiSqrDistanceFull_Norm_8u32f_C1R</a>   | 1893 |
| 7.117.2.15 | <a href="#">nppiSqrDistanceFull_Norm_8u32f_C3R</a>   | 1894 |
| 7.117.2.16 | <a href="#">nppiSqrDistanceFull_Norm_8u32f_C4R</a>   | 1894 |
| 7.117.2.17 | <a href="#">nppiSqrDistanceFull_Norm_8u_AC4RSfs</a>  | 1895 |
| 7.117.2.18 | <a href="#">nppiSqrDistanceFull_Norm_8u_C1RSfs</a>   | 1895 |
| 7.117.2.19 | <a href="#">nppiSqrDistanceFull_Norm_8u_C3RSfs</a>   | 1896 |
| 7.117.2.20 | <a href="#">nppiSqrDistanceFull_Norm_8u_C4RSfs</a>   | 1896 |
| 7.118      | <a href="#">SqrDistanceSame_Norm</a>                 | 1897 |
| 7.118.1    | <a href="#">Detailed Description</a>                 | 1899 |
| 7.118.2    | <a href="#">Function Documentation</a>               | 1899 |
| 7.118.2.1  | <a href="#">nppiSqrDistanceSame_Norm_16u32f_AC4R</a> | 1899 |
| 7.118.2.2  | <a href="#">nppiSqrDistanceSame_Norm_16u32f_C1R</a>  | 1899 |
| 7.118.2.3  | <a href="#">nppiSqrDistanceSame_Norm_16u32f_C3R</a>  | 1900 |
| 7.118.2.4  | <a href="#">nppiSqrDistanceSame_Norm_16u32f_C4R</a>  | 1900 |
| 7.118.2.5  | <a href="#">nppiSqrDistanceSame_Norm_32f_AC4R</a>    | 1901 |
| 7.118.2.6  | <a href="#">nppiSqrDistanceSame_Norm_32f_C1R</a>     | 1901 |
| 7.118.2.7  | <a href="#">nppiSqrDistanceSame_Norm_32f_C3R</a>     | 1901 |
| 7.118.2.8  | <a href="#">nppiSqrDistanceSame_Norm_32f_C4R</a>     | 1902 |
| 7.118.2.9  | <a href="#">nppiSqrDistanceSame_Norm_8s32f_AC4R</a>  | 1902 |
| 7.118.2.10 | <a href="#">nppiSqrDistanceSame_Norm_8s32f_C1R</a>   | 1903 |
| 7.118.2.11 | <a href="#">nppiSqrDistanceSame_Norm_8s32f_C3R</a>   | 1903 |
| 7.118.2.12 | <a href="#">nppiSqrDistanceSame_Norm_8s32f_C4R</a>   | 1904 |
| 7.118.2.13 | <a href="#">nppiSqrDistanceSame_Norm_8u32f_AC4R</a>  | 1904 |
| 7.118.2.14 | <a href="#">nppiSqrDistanceSame_Norm_8u32f_C1R</a>   | 1904 |

|            |   |      |
|------------|---|------|
| 7.118.2.15 | nppiSqrDistanceSame_Norm_8u32f_C3R . . . . .    | 1905 |
| 7.118.2.16 | nppiSqrDistanceSame_Norm_8u32f_C4R . . . . .    | 1905 |
| 7.118.2.17 | nppiSqrDistanceSame_Norm_8u_AC4RSfs . . . . .   | 1906 |
| 7.118.2.18 | nppiSqrDistanceSame_Norm_8u_C1RSfs . . . . .    | 1906 |
| 7.118.2.19 | nppiSqrDistanceSame_Norm_8u_C3RSfs . . . . .    | 1907 |
| 7.118.2.20 | nppiSqrDistanceSame_Norm_8u_C4RSfs . . . . .    | 1907 |
| 7.119      | SqrDistanceValid_Norm . . . . .                 | 1908 |
| 7.119.1    | Detailed Description . . . . .                  | 1910 |
| 7.119.2    | Function Documentation . . . . .                | 1910 |
| 7.119.2.1  | nppiSqrDistanceValid_Norm_16u32f_AC4R . . . . . | 1910 |
| 7.119.2.2  | nppiSqrDistanceValid_Norm_16u32f_C1R . . . . .  | 1910 |
| 7.119.2.3  | nppiSqrDistanceValid_Norm_16u32f_C3R . . . . .  | 1911 |
| 7.119.2.4  | nppiSqrDistanceValid_Norm_16u32f_C4R . . . . .  | 1911 |
| 7.119.2.5  | nppiSqrDistanceValid_Norm_32f_AC4R . . . . .    | 1912 |
| 7.119.2.6  | nppiSqrDistanceValid_Norm_32f_C1R . . . . .     | 1912 |
| 7.119.2.7  | nppiSqrDistanceValid_Norm_32f_C3R . . . . .     | 1912 |
| 7.119.2.8  | nppiSqrDistanceValid_Norm_32f_C4R . . . . .     | 1913 |
| 7.119.2.9  | nppiSqrDistanceValid_Norm_8s32f_AC4R . . . . .  | 1913 |
| 7.119.2.10 | nppiSqrDistanceValid_Norm_8s32f_C1R . . . . .   | 1914 |
| 7.119.2.11 | nppiSqrDistanceValid_Norm_8s32f_C3R . . . . .   | 1914 |
| 7.119.2.12 | nppiSqrDistanceValid_Norm_8s32f_C4R . . . . .   | 1915 |
| 7.119.2.13 | nppiSqrDistanceValid_Norm_8u32f_AC4R . . . . .  | 1915 |
| 7.119.2.14 | nppiSqrDistanceValid_Norm_8u32f_C1R . . . . .   | 1915 |
| 7.119.2.15 | nppiSqrDistanceValid_Norm_8u32f_C3R . . . . .   | 1916 |
| 7.119.2.16 | nppiSqrDistanceValid_Norm_8u32f_C4R . . . . .   | 1916 |
| 7.119.2.17 | nppiSqrDistanceValid_Norm_8u_AC4RSfs . . . . .  | 1917 |
| 7.119.2.18 | nppiSqrDistanceValid_Norm_8u_C1RSfs . . . . .   | 1917 |
| 7.119.2.19 | nppiSqrDistanceValid_Norm_8u_C3RSfs . . . . .   | 1918 |
| 7.119.2.20 | nppiSqrDistanceValid_Norm_8u_C4RSfs . . . . .   | 1918 |
| 7.120      | CrossCorrFull_Norm . . . . .                    | 1919 |
| 7.120.1    | Detailed Description . . . . .                  | 1920 |
| 7.120.2    | Function Documentation . . . . .                | 1921 |
| 7.120.2.1  | nppiCrossCorrFull_Norm_16u32f_AC4R . . . . .    | 1921 |
| 7.120.2.2  | nppiCrossCorrFull_Norm_16u32f_C1R . . . . .     | 1921 |
| 7.120.2.3  | nppiCrossCorrFull_Norm_16u32f_C3R . . . . .     | 1922 |
| 7.120.2.4  | nppiCrossCorrFull_Norm_16u32f_C4R . . . . .     | 1922 |

|  |      |
|--|------|
| 7.120.2.5 nppiCrossCorrFull_Norm_32f_AC4R . . . . .    | 1922 |
| 7.120.2.6 nppiCrossCorrFull_Norm_32f_C1R . . . . .     | 1923 |
| 7.120.2.7 nppiCrossCorrFull_Norm_32f_C3R . . . . .     | 1923 |
| 7.120.2.8 nppiCrossCorrFull_Norm_32f_C4R . . . . .     | 1924 |
| 7.120.2.9 nppiCrossCorrFull_Norm_8s32f_AC4R . . . . .  | 1924 |
| 7.120.2.10 nppiCrossCorrFull_Norm_8s32f_C1R . . . . .  | 1925 |
| 7.120.2.11 nppiCrossCorrFull_Norm_8s32f_C3R . . . . .  | 1925 |
| 7.120.2.12 nppiCrossCorrFull_Norm_8s32f_C4R . . . . .  | 1925 |
| 7.120.2.13 nppiCrossCorrFull_Norm_8u32f_AC4R . . . . . | 1926 |
| 7.120.2.14 nppiCrossCorrFull_Norm_8u32f_C1R . . . . .  | 1926 |
| 7.120.2.15 nppiCrossCorrFull_Norm_8u32f_C3R . . . . .  | 1927 |
| 7.120.2.16 nppiCrossCorrFull_Norm_8u32f_C4R . . . . .  | 1927 |
| 7.120.2.17 nppiCrossCorrFull_Norm_8u_AC4RSfs . . . . . | 1928 |
| 7.120.2.18 nppiCrossCorrFull_Norm_8u_C1RSfs . . . . .  | 1928 |
| 7.120.2.19 nppiCrossCorrFull_Norm_8u_C3RSfs . . . . .  | 1929 |
| 7.120.2.20 nppiCrossCorrFull_Norm_8u_C4RSfs . . . . .  | 1929 |
| 7.121 CrossCorrSame_Norm . . . . .                     | 1930 |
| 7.121.1 Detailed Description . . . . .                 | 1931 |
| 7.121.2 Function Documentation . . . . .               | 1932 |
| 7.121.2.1 nppiCrossCorrSame_Norm_16u32f_AC4R . . . . . | 1932 |
| 7.121.2.2 nppiCrossCorrSame_Norm_16u32f_C1R . . . . .  | 1932 |
| 7.121.2.3 nppiCrossCorrSame_Norm_16u32f_C3R . . . . .  | 1933 |
| 7.121.2.4 nppiCrossCorrSame_Norm_16u32f_C4R . . . . .  | 1933 |
| 7.121.2.5 nppiCrossCorrSame_Norm_32f_AC4R . . . . .    | 1933 |
| 7.121.2.6 nppiCrossCorrSame_Norm_32f_C1R . . . . .     | 1934 |
| 7.121.2.7 nppiCrossCorrSame_Norm_32f_C3R . . . . .     | 1934 |
| 7.121.2.8 nppiCrossCorrSame_Norm_32f_C4R . . . . .     | 1935 |
| 7.121.2.9 nppiCrossCorrSame_Norm_8s32f_AC4R . . . . .  | 1935 |
| 7.121.2.10 nppiCrossCorrSame_Norm_8s32f_C1R . . . . .  | 1936 |
| 7.121.2.11 nppiCrossCorrSame_Norm_8s32f_C3R . . . . .  | 1936 |
| 7.121.2.12 nppiCrossCorrSame_Norm_8s32f_C4R . . . . .  | 1936 |
| 7.121.2.13 nppiCrossCorrSame_Norm_8u32f_AC4R . . . . . | 1937 |
| 7.121.2.14 nppiCrossCorrSame_Norm_8u32f_C1R . . . . .  | 1937 |
| 7.121.2.15 nppiCrossCorrSame_Norm_8u32f_C3R . . . . .  | 1938 |
| 7.121.2.16 nppiCrossCorrSame_Norm_8u32f_C4R . . . . .  | 1938 |
| 7.121.2.17 nppiCrossCorrSame_Norm_8u_AC4RSfs . . . . . | 1939 |

|            |   |      |
|------------|---|------|
| 7.121.2.18 | <a href="#">nppiCrossCorrSame_Norm_8u_C1RSfs</a>    | 1939 |
| 7.121.2.19 | <a href="#">nppiCrossCorrSame_Norm_8u_C3RSfs</a>    | 1940 |
| 7.121.2.20 | <a href="#">nppiCrossCorrSame_Norm_8u_C4RSfs</a>    | 1940 |
| 7.122      | <a href="#">CrossCorrValid_Norm</a>                 | 1941 |
| 7.122.1    | <a href="#">Detailed Description</a>                | 1942 |
| 7.122.2    | <a href="#">Function Documentation</a>              | 1943 |
| 7.122.2.1  | <a href="#">nppiCrossCorrValid_Norm_16u32f_AC4R</a> | 1943 |
| 7.122.2.2  | <a href="#">nppiCrossCorrValid_Norm_16u32f_C1R</a>  | 1943 |
| 7.122.2.3  | <a href="#">nppiCrossCorrValid_Norm_16u32f_C3R</a>  | 1944 |
| 7.122.2.4  | <a href="#">nppiCrossCorrValid_Norm_16u32f_C4R</a>  | 1944 |
| 7.122.2.5  | <a href="#">nppiCrossCorrValid_Norm_32f_AC4R</a>    | 1944 |
| 7.122.2.6  | <a href="#">nppiCrossCorrValid_Norm_32f_C1R</a>     | 1945 |
| 7.122.2.7  | <a href="#">nppiCrossCorrValid_Norm_32f_C3R</a>     | 1945 |
| 7.122.2.8  | <a href="#">nppiCrossCorrValid_Norm_32f_C4R</a>     | 1946 |
| 7.122.2.9  | <a href="#">nppiCrossCorrValid_Norm_8s32f_AC4R</a>  | 1946 |
| 7.122.2.10 | <a href="#">nppiCrossCorrValid_Norm_8s32f_C1R</a>   | 1947 |
| 7.122.2.11 | <a href="#">nppiCrossCorrValid_Norm_8s32f_C3R</a>   | 1947 |
| 7.122.2.12 | <a href="#">nppiCrossCorrValid_Norm_8s32f_C4R</a>   | 1947 |
| 7.122.2.13 | <a href="#">nppiCrossCorrValid_Norm_8u32f_AC4R</a>  | 1948 |
| 7.122.2.14 | <a href="#">nppiCrossCorrValid_Norm_8u32f_C1R</a>   | 1948 |
| 7.122.2.15 | <a href="#">nppiCrossCorrValid_Norm_8u32f_C3R</a>   | 1949 |
| 7.122.2.16 | <a href="#">nppiCrossCorrValid_Norm_8u32f_C4R</a>   | 1949 |
| 7.122.2.17 | <a href="#">nppiCrossCorrValid_Norm_8u_AC4RSfs</a>  | 1950 |
| 7.122.2.18 | <a href="#">nppiCrossCorrValid_Norm_8u_C1RSfs</a>   | 1950 |
| 7.122.2.19 | <a href="#">nppiCrossCorrValid_Norm_8u_C3RSfs</a>   | 1951 |
| 7.122.2.20 | <a href="#">nppiCrossCorrValid_Norm_8u_C4RSfs</a>   | 1951 |
| 7.123      | <a href="#">CrossCorrValid</a>                      | 1952 |
| 7.123.1    | <a href="#">Detailed Description</a>                | 1952 |
| 7.123.2    | <a href="#">Function Documentation</a>              | 1952 |
| 7.123.2.1  | <a href="#">nppiCrossCorrValid_16u32f_C1R</a>       | 1952 |
| 7.123.2.2  | <a href="#">nppiCrossCorrValid_32f_C1R</a>          | 1953 |
| 7.123.2.3  | <a href="#">nppiCrossCorrValid_8s32f_C1R</a>        | 1953 |
| 7.123.2.4  | <a href="#">nppiCrossCorrValid_8u32f_C1R</a>        | 1954 |
| 7.124      | <a href="#">CrossCorrFull_NormLevel</a>             | 1955 |
| 7.124.1    | <a href="#">Detailed Description</a>                | 1958 |
| 7.124.2    | <a href="#">Function Documentation</a>              | 1959 |

|            |  |      |
|------------|--|------|
| 7.124.2.1  | <a href="#">nppiCrossCorrFull_NormLevel_16u32f_AC4R</a>        | 1959 |
| 7.124.2.2  | <a href="#">nppiCrossCorrFull_NormLevel_16u32f_C1R</a>         | 1959 |
| 7.124.2.3  | <a href="#">nppiCrossCorrFull_NormLevel_16u32f_C3R</a>         | 1960 |
| 7.124.2.4  | <a href="#">nppiCrossCorrFull_NormLevel_16u32f_C4R</a>         | 1960 |
| 7.124.2.5  | <a href="#">nppiCrossCorrFull_NormLevel_32f_AC4R</a>           | 1961 |
| 7.124.2.6  | <a href="#">nppiCrossCorrFull_NormLevel_32f_C1R</a>            | 1961 |
| 7.124.2.7  | <a href="#">nppiCrossCorrFull_NormLevel_32f_C3R</a>            | 1962 |
| 7.124.2.8  | <a href="#">nppiCrossCorrFull_NormLevel_32f_C4R</a>            | 1962 |
| 7.124.2.9  | <a href="#">nppiCrossCorrFull_NormLevel_8s32f_AC4R</a>         | 1963 |
| 7.124.2.10 | <a href="#">nppiCrossCorrFull_NormLevel_8s32f_C1R</a>          | 1963 |
| 7.124.2.11 | <a href="#">nppiCrossCorrFull_NormLevel_8s32f_C3R</a>          | 1964 |
| 7.124.2.12 | <a href="#">nppiCrossCorrFull_NormLevel_8s32f_C4R</a>          | 1964 |
| 7.124.2.13 | <a href="#">nppiCrossCorrFull_NormLevel_8u32f_AC4R</a>         | 1965 |
| 7.124.2.14 | <a href="#">nppiCrossCorrFull_NormLevel_8u32f_C1R</a>          | 1965 |
| 7.124.2.15 | <a href="#">nppiCrossCorrFull_NormLevel_8u32f_C3R</a>          | 1966 |
| 7.124.2.16 | <a href="#">nppiCrossCorrFull_NormLevel_8u32f_C4R</a>          | 1966 |
| 7.124.2.17 | <a href="#">nppiCrossCorrFull_NormLevel_8u_AC4RSfs</a>         | 1967 |
| 7.124.2.18 | <a href="#">nppiCrossCorrFull_NormLevel_8u_C1RSfs</a>          | 1967 |
| 7.124.2.19 | <a href="#">nppiCrossCorrFull_NormLevel_8u_C3RSfs</a>          | 1968 |
| 7.124.2.20 | <a href="#">nppiCrossCorrFull_NormLevel_8u_C4RSfs</a>          | 1968 |
| 7.124.2.21 | <a href="#">nppiFullNormLevelGetBufferHostSize_16u32f_AC4R</a> | 1969 |
| 7.124.2.22 | <a href="#">nppiFullNormLevelGetBufferHostSize_16u32f_C1R</a>  | 1969 |
| 7.124.2.23 | <a href="#">nppiFullNormLevelGetBufferHostSize_16u32f_C3R</a>  | 1969 |
| 7.124.2.24 | <a href="#">nppiFullNormLevelGetBufferHostSize_16u32f_C4R</a>  | 1969 |
| 7.124.2.25 | <a href="#">nppiFullNormLevelGetBufferHostSize_32f_AC4R</a>    | 1970 |
| 7.124.2.26 | <a href="#">nppiFullNormLevelGetBufferHostSize_32f_C1R</a>     | 1970 |
| 7.124.2.27 | <a href="#">nppiFullNormLevelGetBufferHostSize_32f_C3R</a>     | 1970 |
| 7.124.2.28 | <a href="#">nppiFullNormLevelGetBufferHostSize_32f_C4R</a>     | 1971 |
| 7.124.2.29 | <a href="#">nppiFullNormLevelGetBufferHostSize_8s32f_AC4R</a>  | 1971 |
| 7.124.2.30 | <a href="#">nppiFullNormLevelGetBufferHostSize_8s32f_C1R</a>   | 1971 |
| 7.124.2.31 | <a href="#">nppiFullNormLevelGetBufferHostSize_8s32f_C3R</a>   | 1971 |
| 7.124.2.32 | <a href="#">nppiFullNormLevelGetBufferHostSize_8s32f_C4R</a>   | 1972 |
| 7.124.2.33 | <a href="#">nppiFullNormLevelGetBufferHostSize_8u32f_AC4R</a>  | 1972 |
| 7.124.2.34 | <a href="#">nppiFullNormLevelGetBufferHostSize_8u32f_C1R</a>   | 1972 |
| 7.124.2.35 | <a href="#">nppiFullNormLevelGetBufferHostSize_8u32f_C3R</a>   | 1973 |
| 7.124.2.36 | <a href="#">nppiFullNormLevelGetBufferHostSize_8u32f_C4R</a>   | 1973 |

|            |  |      |
|------------|--|------|
| 7.124.2.37 | <a href="#">ippiFullNormLevelGetBufferHostSize_8u_AC4RSfs</a>  | 1973 |
| 7.124.2.38 | <a href="#">ippiFullNormLevelGetBufferHostSize_8u_C1RSfs</a>   | 1973 |
| 7.124.2.39 | <a href="#">ippiFullNormLevelGetBufferHostSize_8u_C3RSfs</a>   | 1974 |
| 7.124.2.40 | <a href="#">ippiFullNormLevelGetBufferHostSize_8u_C4RSfs</a>   | 1974 |
| 7.125      | <a href="#">CrossCorrSame_NormLevel</a>                        | 1975 |
| 7.125.1    | <a href="#">Detailed Description</a>                           | 1978 |
| 7.125.2    | <a href="#">Function Documentation</a>                         | 1979 |
| 7.125.2.1  | <a href="#">ippiCrossCorrSame_NormLevel_16u32f_AC4R</a>        | 1979 |
| 7.125.2.2  | <a href="#">ippiCrossCorrSame_NormLevel_16u32f_C1R</a>         | 1979 |
| 7.125.2.3  | <a href="#">ippiCrossCorrSame_NormLevel_16u32f_C3R</a>         | 1980 |
| 7.125.2.4  | <a href="#">ippiCrossCorrSame_NormLevel_16u32f_C4R</a>         | 1980 |
| 7.125.2.5  | <a href="#">ippiCrossCorrSame_NormLevel_32f_AC4R</a>           | 1981 |
| 7.125.2.6  | <a href="#">ippiCrossCorrSame_NormLevel_32f_C1R</a>            | 1981 |
| 7.125.2.7  | <a href="#">ippiCrossCorrSame_NormLevel_32f_C3R</a>            | 1982 |
| 7.125.2.8  | <a href="#">ippiCrossCorrSame_NormLevel_32f_C4R</a>            | 1982 |
| 7.125.2.9  | <a href="#">ippiCrossCorrSame_NormLevel_8s32f_AC4R</a>         | 1983 |
| 7.125.2.10 | <a href="#">ippiCrossCorrSame_NormLevel_8s32f_C1R</a>          | 1983 |
| 7.125.2.11 | <a href="#">ippiCrossCorrSame_NormLevel_8s32f_C3R</a>          | 1984 |
| 7.125.2.12 | <a href="#">ippiCrossCorrSame_NormLevel_8s32f_C4R</a>          | 1984 |
| 7.125.2.13 | <a href="#">ippiCrossCorrSame_NormLevel_8u32f_AC4R</a>         | 1985 |
| 7.125.2.14 | <a href="#">ippiCrossCorrSame_NormLevel_8u32f_C1R</a>          | 1985 |
| 7.125.2.15 | <a href="#">ippiCrossCorrSame_NormLevel_8u32f_C3R</a>          | 1986 |
| 7.125.2.16 | <a href="#">ippiCrossCorrSame_NormLevel_8u32f_C4R</a>          | 1986 |
| 7.125.2.17 | <a href="#">ippiCrossCorrSame_NormLevel_8u_AC4RSfs</a>         | 1987 |
| 7.125.2.18 | <a href="#">ippiCrossCorrSame_NormLevel_8u_C1RSfs</a>          | 1987 |
| 7.125.2.19 | <a href="#">ippiCrossCorrSame_NormLevel_8u_C3RSfs</a>          | 1988 |
| 7.125.2.20 | <a href="#">ippiCrossCorrSame_NormLevel_8u_C4RSfs</a>          | 1988 |
| 7.125.2.21 | <a href="#">ippiSameNormLevelGetBufferHostSize_16u32f_AC4R</a> | 1989 |
| 7.125.2.22 | <a href="#">ippiSameNormLevelGetBufferHostSize_16u32f_C1R</a>  | 1989 |
| 7.125.2.23 | <a href="#">ippiSameNormLevelGetBufferHostSize_16u32f_C3R</a>  | 1989 |
| 7.125.2.24 | <a href="#">ippiSameNormLevelGetBufferHostSize_16u32f_C4R</a>  | 1989 |
| 7.125.2.25 | <a href="#">ippiSameNormLevelGetBufferHostSize_32f_AC4R</a>    | 1990 |
| 7.125.2.26 | <a href="#">ippiSameNormLevelGetBufferHostSize_32f_C1R</a>     | 1990 |
| 7.125.2.27 | <a href="#">ippiSameNormLevelGetBufferHostSize_32f_C3R</a>     | 1990 |
| 7.125.2.28 | <a href="#">ippiSameNormLevelGetBufferHostSize_32f_C4R</a>     | 1991 |
| 7.125.2.29 | <a href="#">ippiSameNormLevelGetBufferHostSize_8s32f_AC4R</a>  | 1991 |

|            |   |      |
|------------|---|------|
| 7.125.2.30 | <a href="#">nppiSameNormLevelGetBufferHostSize_8s32f_C1R</a>    | 1991 |
| 7.125.2.31 | <a href="#">nppiSameNormLevelGetBufferHostSize_8s32f_C3R</a>    | 1991 |
| 7.125.2.32 | <a href="#">nppiSameNormLevelGetBufferHostSize_8s32f_C4R</a>    | 1992 |
| 7.125.2.33 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u32f_AC4R</a>   | 1992 |
| 7.125.2.34 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u32f_C1R</a>    | 1992 |
| 7.125.2.35 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u32f_C3R</a>    | 1993 |
| 7.125.2.36 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u32f_C4R</a>    | 1993 |
| 7.125.2.37 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u_AC4RSfs</a>   | 1993 |
| 7.125.2.38 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u_C1RSfs</a>    | 1993 |
| 7.125.2.39 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u_C3RSfs</a>    | 1994 |
| 7.125.2.40 | <a href="#">nppiSameNormLevelGetBufferHostSize_8u_C4RSfs</a>    | 1994 |
| 7.126      | <a href="#">CrossCorrValid_NormLevel</a>                        | 1995 |
| 7.126.1    | <a href="#">Detailed Description</a>                            | 1998 |
| 7.126.2    | <a href="#">Function Documentation</a>                          | 1999 |
| 7.126.2.1  | <a href="#">nppiCrossCorrValid_NormLevel_16u32f_AC4R</a>        | 1999 |
| 7.126.2.2  | <a href="#">nppiCrossCorrValid_NormLevel_16u32f_C1R</a>         | 1999 |
| 7.126.2.3  | <a href="#">nppiCrossCorrValid_NormLevel_16u32f_C3R</a>         | 2000 |
| 7.126.2.4  | <a href="#">nppiCrossCorrValid_NormLevel_16u32f_C4R</a>         | 2000 |
| 7.126.2.5  | <a href="#">nppiCrossCorrValid_NormLevel_32f_AC4R</a>           | 2001 |
| 7.126.2.6  | <a href="#">nppiCrossCorrValid_NormLevel_32f_C1R</a>            | 2001 |
| 7.126.2.7  | <a href="#">nppiCrossCorrValid_NormLevel_32f_C3R</a>            | 2002 |
| 7.126.2.8  | <a href="#">nppiCrossCorrValid_NormLevel_32f_C4R</a>            | 2002 |
| 7.126.2.9  | <a href="#">nppiCrossCorrValid_NormLevel_8s32f_AC4R</a>         | 2003 |
| 7.126.2.10 | <a href="#">nppiCrossCorrValid_NormLevel_8s32f_C1R</a>          | 2003 |
| 7.126.2.11 | <a href="#">nppiCrossCorrValid_NormLevel_8s32f_C3R</a>          | 2004 |
| 7.126.2.12 | <a href="#">nppiCrossCorrValid_NormLevel_8s32f_C4R</a>          | 2004 |
| 7.126.2.13 | <a href="#">nppiCrossCorrValid_NormLevel_8u32f_AC4R</a>         | 2005 |
| 7.126.2.14 | <a href="#">nppiCrossCorrValid_NormLevel_8u32f_C1R</a>          | 2005 |
| 7.126.2.15 | <a href="#">nppiCrossCorrValid_NormLevel_8u32f_C3R</a>          | 2006 |
| 7.126.2.16 | <a href="#">nppiCrossCorrValid_NormLevel_8u32f_C4R</a>          | 2006 |
| 7.126.2.17 | <a href="#">nppiCrossCorrValid_NormLevel_8u_AC4RSfs</a>         | 2007 |
| 7.126.2.18 | <a href="#">nppiCrossCorrValid_NormLevel_8u_C1RSfs</a>          | 2007 |
| 7.126.2.19 | <a href="#">nppiCrossCorrValid_NormLevel_8u_C3RSfs</a>          | 2008 |
| 7.126.2.20 | <a href="#">nppiCrossCorrValid_NormLevel_8u_C4RSfs</a>          | 2008 |
| 7.126.2.21 | <a href="#">nppiValidNormLevelGetBufferHostSize_16u32f_AC4R</a> | 2009 |
| 7.126.2.22 | <a href="#">nppiValidNormLevelGetBufferHostSize_16u32f_C1R</a>  | 2009 |

|            |  |      |
|------------|--|------|
| 7.126.2.23 | <a href="#">nppiValidNormLevelGetBufferHostSize_16u32f_C3R</a> | 2009 |
| 7.126.2.24 | <a href="#">nppiValidNormLevelGetBufferHostSize_16u32f_C4R</a> | 2009 |
| 7.126.2.25 | <a href="#">nppiValidNormLevelGetBufferHostSize_32f_AC4R</a>   | 2010 |
| 7.126.2.26 | <a href="#">nppiValidNormLevelGetBufferHostSize_32f_C1R</a>    | 2010 |
| 7.126.2.27 | <a href="#">nppiValidNormLevelGetBufferHostSize_32f_C3R</a>    | 2010 |
| 7.126.2.28 | <a href="#">nppiValidNormLevelGetBufferHostSize_32f_C4R</a>    | 2011 |
| 7.126.2.29 | <a href="#">nppiValidNormLevelGetBufferHostSize_8s32f_AC4R</a> | 2011 |
| 7.126.2.30 | <a href="#">nppiValidNormLevelGetBufferHostSize_8s32f_C1R</a>  | 2011 |
| 7.126.2.31 | <a href="#">nppiValidNormLevelGetBufferHostSize_8s32f_C3R</a>  | 2011 |
| 7.126.2.32 | <a href="#">nppiValidNormLevelGetBufferHostSize_8s32f_C4R</a>  | 2012 |
| 7.126.2.33 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u32f_AC4R</a> | 2012 |
| 7.126.2.34 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u32f_C1R</a>  | 2012 |
| 7.126.2.35 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u32f_C3R</a>  | 2013 |
| 7.126.2.36 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u32f_C4R</a>  | 2013 |
| 7.126.2.37 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u_AC4RSfs</a> | 2013 |
| 7.126.2.38 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u_C1RSfs</a>  | 2013 |
| 7.126.2.39 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u_C3RSfs</a>  | 2014 |
| 7.126.2.40 | <a href="#">nppiValidNormLevelGetBufferHostSize_8u_C4RSfs</a>  | 2014 |
| 7.127      | <a href="#">Image Quality Index</a>                            | 2015 |
| 7.127.1    | <a href="#">Detailed Description</a>                           | 2017 |
| 7.127.2    | <a href="#">Function Documentation</a>                         | 2017 |
| 7.127.2.1  | <a href="#">nppiQualityIndex_16u32f_AC4R</a>                   | 2017 |
| 7.127.2.2  | <a href="#">nppiQualityIndex_16u32f_C1R</a>                    | 2017 |
| 7.127.2.3  | <a href="#">nppiQualityIndex_16u32f_C3R</a>                    | 2018 |
| 7.127.2.4  | <a href="#">nppiQualityIndex_32f_AC4R</a>                      | 2018 |
| 7.127.2.5  | <a href="#">nppiQualityIndex_32f_C1R</a>                       | 2019 |
| 7.127.2.6  | <a href="#">nppiQualityIndex_32f_C3R</a>                       | 2019 |
| 7.127.2.7  | <a href="#">nppiQualityIndex_8u32f_AC4R</a>                    | 2020 |
| 7.127.2.8  | <a href="#">nppiQualityIndex_8u32f_C1R</a>                     | 2020 |
| 7.127.2.9  | <a href="#">nppiQualityIndex_8u32f_C3R</a>                     | 2020 |
| 7.127.2.10 | <a href="#">nppiQualityIndexGetBufferHostSize_16u32f_AC4R</a>  | 2021 |
| 7.127.2.11 | <a href="#">nppiQualityIndexGetBufferHostSize_16u32f_C1R</a>   | 2021 |
| 7.127.2.12 | <a href="#">nppiQualityIndexGetBufferHostSize_16u32f_C3R</a>   | 2022 |
| 7.127.2.13 | <a href="#">nppiQualityIndexGetBufferHostSize_32f_AC4R</a>     | 2022 |
| 7.127.2.14 | <a href="#">nppiQualityIndexGetBufferHostSize_32f_C1R</a>      | 2022 |
| 7.127.2.15 | <a href="#">nppiQualityIndexGetBufferHostSize_32f_C3R</a>      | 2022 |



|            |  |      |
|------------|--|------|
| 7.127.2.16 | <a href="#">nppiQualityIndexGetBufferHostSize_8u32f_AC4R</a> | 2023 |
| 7.127.2.17 | <a href="#">nppiQualityIndexGetBufferHostSize_8u32f_C1R</a>  | 2023 |
| 7.127.2.18 | <a href="#">nppiQualityIndexGetBufferHostSize_8u32f_C3R</a>  | 2023 |
| 7.128      | <a href="#">MaximumError</a>                                 | 2024 |
| 7.128.1    | <a href="#">Detailed Description</a>                         | 2027 |
| 7.128.2    | <a href="#">Function Documentation</a>                       | 2027 |
| 7.128.2.1  | <a href="#">nppiMaximumError_16s_C1R</a>                     | 2027 |
| 7.128.2.2  | <a href="#">nppiMaximumError_16s_C2R</a>                     | 2028 |
| 7.128.2.3  | <a href="#">nppiMaximumError_16s_C3R</a>                     | 2028 |
| 7.128.2.4  | <a href="#">nppiMaximumError_16s_C4R</a>                     | 2029 |
| 7.128.2.5  | <a href="#">nppiMaximumError_16sc_C1R</a>                    | 2029 |
| 7.128.2.6  | <a href="#">nppiMaximumError_16sc_C2R</a>                    | 2029 |
| 7.128.2.7  | <a href="#">nppiMaximumError_16sc_C3R</a>                    | 2030 |
| 7.128.2.8  | <a href="#">nppiMaximumError_16sc_C4R</a>                    | 2030 |
| 7.128.2.9  | <a href="#">nppiMaximumError_16u_C1R</a>                     | 2031 |
| 7.128.2.10 | <a href="#">nppiMaximumError_16u_C2R</a>                     | 2031 |
| 7.128.2.11 | <a href="#">nppiMaximumError_16u_C3R</a>                     | 2032 |
| 7.128.2.12 | <a href="#">nppiMaximumError_16u_C4R</a>                     | 2032 |
| 7.128.2.13 | <a href="#">nppiMaximumError_32f_C1R</a>                     | 2032 |
| 7.128.2.14 | <a href="#">nppiMaximumError_32f_C2R</a>                     | 2033 |
| 7.128.2.15 | <a href="#">nppiMaximumError_32f_C3R</a>                     | 2033 |
| 7.128.2.16 | <a href="#">nppiMaximumError_32f_C4R</a>                     | 2034 |
| 7.128.2.17 | <a href="#">nppiMaximumError_32fc_C1R</a>                    | 2034 |
| 7.128.2.18 | <a href="#">nppiMaximumError_32fc_C2R</a>                    | 2035 |
| 7.128.2.19 | <a href="#">nppiMaximumError_32fc_C3R</a>                    | 2035 |
| 7.128.2.20 | <a href="#">nppiMaximumError_32fc_C4R</a>                    | 2036 |
| 7.128.2.21 | <a href="#">nppiMaximumError_32s_C1R</a>                     | 2036 |
| 7.128.2.22 | <a href="#">nppiMaximumError_32s_C2R</a>                     | 2036 |
| 7.128.2.23 | <a href="#">nppiMaximumError_32s_C3R</a>                     | 2037 |
| 7.128.2.24 | <a href="#">nppiMaximumError_32s_C4R</a>                     | 2037 |
| 7.128.2.25 | <a href="#">nppiMaximumError_32sc_C1R</a>                    | 2038 |
| 7.128.2.26 | <a href="#">nppiMaximumError_32sc_C2R</a>                    | 2038 |
| 7.128.2.27 | <a href="#">nppiMaximumError_32sc_C3R</a>                    | 2039 |
| 7.128.2.28 | <a href="#">nppiMaximumError_32sc_C4R</a>                    | 2039 |
| 7.128.2.29 | <a href="#">nppiMaximumError_32u_C1R</a>                     | 2039 |
| 7.128.2.30 | <a href="#">nppiMaximumError_32u_C2R</a>                     | 2040 |

|            |   |      |
|------------|---|------|
| 7.128.2.3  | <a href="#">ippiMaximumError_32u_C3R</a>  | 2040 |
| 7.128.2.32 | <a href="#">ippiMaximumError_32u_C4R</a>  | 2041 |
| 7.128.2.33 | <a href="#">ippiMaximumError_64f_C1R</a>  | 2041 |
| 7.128.2.34 | <a href="#">ippiMaximumError_64f_C2R</a>  | 2042 |
| 7.128.2.35 | <a href="#">ippiMaximumError_64f_C3R</a>  | 2042 |
| 7.128.2.36 | <a href="#">ippiMaximumError_64f_C4R</a>  | 2042 |
| 7.128.2.37 | <a href="#">ippiMaximumError_8s_C1R</a>   | 2043 |
| 7.128.2.38 | <a href="#">ippiMaximumError_8s_C2R</a>   | 2043 |
| 7.128.2.39 | <a href="#">ippiMaximumError_8s_C3R</a>   | 2044 |
| 7.128.2.40 | <a href="#">ippiMaximumError_8s_C4R</a>   | 2044 |
| 7.128.2.41 | <a href="#">ippiMaximumError_8u_C1R</a>   | 2045 |
| 7.128.2.42 | <a href="#">ippiMaximumError_8u_C2R</a>   | 2045 |
| 7.128.2.43 | <a href="#">ippiMaximumError_8u_C3R</a>   | 2045 |
| 7.128.2.44 | <a href="#">ippiMaximumError_8u_C4R</a>   | 2046 |
| 7.129      | <a href="#">AverageError</a>              | 2047 |
| 7.129.1    | <a href="#">Detailed Description</a>      | 2050 |
| 7.129.2    | <a href="#">Function Documentation</a>    | 2050 |
| 7.129.2.1  | <a href="#">ippiAverageError_16s_C1R</a>  | 2050 |
| 7.129.2.2  | <a href="#">ippiAverageError_16s_C2R</a>  | 2051 |
| 7.129.2.3  | <a href="#">ippiAverageError_16s_C3R</a>  | 2051 |
| 7.129.2.4  | <a href="#">ippiAverageError_16s_C4R</a>  | 2052 |
| 7.129.2.5  | <a href="#">ippiAverageError_16sc_C1R</a> | 2052 |
| 7.129.2.6  | <a href="#">ippiAverageError_16sc_C2R</a> | 2053 |
| 7.129.2.7  | <a href="#">ippiAverageError_16sc_C3R</a> | 2053 |
| 7.129.2.8  | <a href="#">ippiAverageError_16sc_C4R</a> | 2053 |
| 7.129.2.9  | <a href="#">ippiAverageError_16u_C1R</a>  | 2054 |
| 7.129.2.10 | <a href="#">ippiAverageError_16u_C2R</a>  | 2054 |
| 7.129.2.11 | <a href="#">ippiAverageError_16u_C3R</a>  | 2055 |
| 7.129.2.12 | <a href="#">ippiAverageError_16u_C4R</a>  | 2055 |
| 7.129.2.13 | <a href="#">ippiAverageError_32f_C1R</a>  | 2056 |
| 7.129.2.14 | <a href="#">ippiAverageError_32f_C2R</a>  | 2056 |
| 7.129.2.15 | <a href="#">ippiAverageError_32f_C3R</a>  | 2056 |
| 7.129.2.16 | <a href="#">ippiAverageError_32f_C4R</a>  | 2057 |
| 7.129.2.17 | <a href="#">ippiAverageError_32fc_C1R</a> | 2057 |
| 7.129.2.18 | <a href="#">ippiAverageError_32fc_C2R</a> | 2058 |
| 7.129.2.19 | <a href="#">ippiAverageError_32fc_C3R</a> | 2058 |

|            |                                   |      |
|------------|-----------------------------------|------|
| 7.129.2.20 | nppiAverageError_32fc_C4R         | 2059 |
| 7.129.2.21 | nppiAverageError_32s_C1R          | 2059 |
| 7.129.2.22 | nppiAverageError_32s_C2R          | 2060 |
| 7.129.2.23 | nppiAverageError_32s_C3R          | 2060 |
| 7.129.2.24 | nppiAverageError_32s_C4R          | 2060 |
| 7.129.2.25 | nppiAverageError_32sc_C1R         | 2061 |
| 7.129.2.26 | nppiAverageError_32sc_C2R         | 2061 |
| 7.129.2.27 | nppiAverageError_32sc_C3R         | 2062 |
| 7.129.2.28 | nppiAverageError_32sc_C4R         | 2062 |
| 7.129.2.29 | nppiAverageError_32u_C1R          | 2063 |
| 7.129.2.30 | nppiAverageError_32u_C2R          | 2063 |
| 7.129.2.31 | nppiAverageError_32u_C3R          | 2063 |
| 7.129.2.32 | nppiAverageError_32u_C4R          | 2064 |
| 7.129.2.33 | nppiAverageError_64f_C1R          | 2064 |
| 7.129.2.34 | nppiAverageError_64f_C2R          | 2065 |
| 7.129.2.35 | nppiAverageError_64f_C3R          | 2065 |
| 7.129.2.36 | nppiAverageError_64f_C4R          | 2066 |
| 7.129.2.37 | nppiAverageError_8s_C1R           | 2066 |
| 7.129.2.38 | nppiAverageError_8s_C2R           | 2067 |
| 7.129.2.39 | nppiAverageError_8s_C3R           | 2067 |
| 7.129.2.40 | nppiAverageError_8s_C4R           | 2067 |
| 7.129.2.41 | nppiAverageError_8u_C1R           | 2068 |
| 7.129.2.42 | nppiAverageError_8u_C2R           | 2068 |
| 7.129.2.43 | nppiAverageError_8u_C3R           | 2069 |
| 7.129.2.44 | nppiAverageError_8u_C4R           | 2069 |
| 7.130      | MaximumRelativeError              | 2070 |
| 7.130.1    | Detailed Description              | 2073 |
| 7.130.2    | Function Documentation            | 2073 |
| 7.130.2.1  | nppiMaximumRelativeError_16s_C1R  | 2073 |
| 7.130.2.2  | nppiMaximumRelativeError_16s_C2R  | 2074 |
| 7.130.2.3  | nppiMaximumRelativeError_16s_C3R  | 2074 |
| 7.130.2.4  | nppiMaximumRelativeError_16s_C4R  | 2075 |
| 7.130.2.5  | nppiMaximumRelativeError_16sc_C1R | 2075 |
| 7.130.2.6  | nppiMaximumRelativeError_16sc_C2R | 2076 |
| 7.130.2.7  | nppiMaximumRelativeError_16sc_C3R | 2076 |
| 7.130.2.8  | nppiMaximumRelativeError_16sc_C4R | 2077 |

|  |      |
|--|------|
| 7.130.2.9 nppiMaximumRelativeError_16u_C1R . . . . .   | 2077 |
| 7.130.2.10 nppiMaximumRelativeError_16u_C2R . . . . .  | 2077 |
| 7.130.2.11 nppiMaximumRelativeError_16u_C3R . . . . .  | 2078 |
| 7.130.2.12 nppiMaximumRelativeError_16u_C4R . . . . .  | 2078 |
| 7.130.2.13 nppiMaximumRelativeError_32f_C1R . . . . .  | 2079 |
| 7.130.2.14 nppiMaximumRelativeError_32f_C2R . . . . .  | 2079 |
| 7.130.2.15 nppiMaximumRelativeError_32f_C3R . . . . .  | 2080 |
| 7.130.2.16 nppiMaximumRelativeError_32f_C4R . . . . .  | 2080 |
| 7.130.2.17 nppiMaximumRelativeError_32fc_C1R . . . . . | 2081 |
| 7.130.2.18 nppiMaximumRelativeError_32fc_C2R . . . . . | 2081 |
| 7.130.2.19 nppiMaximumRelativeError_32fc_C3R . . . . . | 2082 |
| 7.130.2.20 nppiMaximumRelativeError_32fc_C4R . . . . . | 2082 |
| 7.130.2.21 nppiMaximumRelativeError_32s_C1R . . . . .  | 2083 |
| 7.130.2.22 nppiMaximumRelativeError_32s_C2R . . . . .  | 2083 |
| 7.130.2.23 nppiMaximumRelativeError_32s_C3R . . . . .  | 2083 |
| 7.130.2.24 nppiMaximumRelativeError_32s_C4R . . . . .  | 2084 |
| 7.130.2.25 nppiMaximumRelativeError_32sc_C1R . . . . . | 2084 |
| 7.130.2.26 nppiMaximumRelativeError_32sc_C2R . . . . . | 2085 |
| 7.130.2.27 nppiMaximumRelativeError_32sc_C3R . . . . . | 2085 |
| 7.130.2.28 nppiMaximumRelativeError_32sc_C4R . . . . . | 2086 |
| 7.130.2.29 nppiMaximumRelativeError_32u_C1R . . . . .  | 2086 |
| 7.130.2.30 nppiMaximumRelativeError_32u_C2R . . . . .  | 2087 |
| 7.130.2.31 nppiMaximumRelativeError_32u_C3R . . . . .  | 2087 |
| 7.130.2.32 nppiMaximumRelativeError_32u_C4R . . . . .  | 2087 |
| 7.130.2.33 nppiMaximumRelativeError_64f_C1R . . . . .  | 2088 |
| 7.130.2.34 nppiMaximumRelativeError_64f_C2R . . . . .  | 2088 |
| 7.130.2.35 nppiMaximumRelativeError_64f_C3R . . . . .  | 2089 |
| 7.130.2.36 nppiMaximumRelativeError_64f_C4R . . . . .  | 2089 |
| 7.130.2.37 nppiMaximumRelativeError_8s_C1R . . . . .   | 2090 |
| 7.130.2.38 nppiMaximumRelativeError_8s_C2R . . . . .   | 2090 |
| 7.130.2.39 nppiMaximumRelativeError_8s_C3R . . . . .   | 2091 |
| 7.130.2.40 nppiMaximumRelativeError_8s_C4R . . . . .   | 2091 |
| 7.130.2.41 nppiMaximumRelativeError_8u_C1R . . . . .   | 2092 |
| 7.130.2.42 nppiMaximumRelativeError_8u_C2R . . . . .   | 2092 |
| 7.130.2.43 nppiMaximumRelativeError_8u_C3R . . . . .   | 2092 |
| 7.130.2.44 nppiMaximumRelativeError_8u_C4R . . . . .   | 2093 |

|  |      |
|--|------|
| 7.131 AverageRelativeError . . . . .                   | 2094 |
| 7.131.1 Detailed Description . . . . .                 | 2097 |
| 7.131.2 Function Documentation . . . . .               | 2097 |
| 7.131.2.1 npplAverageRelativeError_16s_C1R . . . . .   | 2097 |
| 7.131.2.2 npplAverageRelativeError_16s_C2R . . . . .   | 2098 |
| 7.131.2.3 npplAverageRelativeError_16s_C3R . . . . .   | 2098 |
| 7.131.2.4 npplAverageRelativeError_16s_C4R . . . . .   | 2099 |
| 7.131.2.5 npplAverageRelativeError_16sc_C1R . . . . .  | 2099 |
| 7.131.2.6 npplAverageRelativeError_16sc_C2R . . . . .  | 2100 |
| 7.131.2.7 npplAverageRelativeError_16sc_C3R . . . . .  | 2100 |
| 7.131.2.8 npplAverageRelativeError_16sc_C4R . . . . .  | 2101 |
| 7.131.2.9 npplAverageRelativeError_16u_C1R . . . . .   | 2101 |
| 7.131.2.10 npplAverageRelativeError_16u_C2R . . . . .  | 2101 |
| 7.131.2.11 npplAverageRelativeError_16u_C3R . . . . .  | 2102 |
| 7.131.2.12 npplAverageRelativeError_16u_C4R . . . . .  | 2102 |
| 7.131.2.13 npplAverageRelativeError_32f_C1R . . . . .  | 2103 |
| 7.131.2.14 npplAverageRelativeError_32f_C2R . . . . .  | 2103 |
| 7.131.2.15 npplAverageRelativeError_32f_C3R . . . . .  | 2104 |
| 7.131.2.16 npplAverageRelativeError_32f_C4R . . . . .  | 2104 |
| 7.131.2.17 npplAverageRelativeError_32fc_C1R . . . . . | 2105 |
| 7.131.2.18 npplAverageRelativeError_32fc_C2R . . . . . | 2105 |
| 7.131.2.19 npplAverageRelativeError_32fc_C3R . . . . . | 2106 |
| 7.131.2.20 npplAverageRelativeError_32fc_C4R . . . . . | 2106 |
| 7.131.2.21 npplAverageRelativeError_32s_C1R . . . . .  | 2107 |
| 7.131.2.22 npplAverageRelativeError_32s_C2R . . . . .  | 2107 |
| 7.131.2.23 npplAverageRelativeError_32s_C3R . . . . .  | 2107 |
| 7.131.2.24 npplAverageRelativeError_32s_C4R . . . . .  | 2108 |
| 7.131.2.25 npplAverageRelativeError_32sc_C1R . . . . . | 2108 |
| 7.131.2.26 npplAverageRelativeError_32sc_C2R . . . . . | 2109 |
| 7.131.2.27 npplAverageRelativeError_32sc_C3R . . . . . | 2109 |
| 7.131.2.28 npplAverageRelativeError_32sc_C4R . . . . . | 2110 |
| 7.131.2.29 npplAverageRelativeError_32u_C1R . . . . .  | 2110 |
| 7.131.2.30 npplAverageRelativeError_32u_C2R . . . . .  | 2111 |
| 7.131.2.31 npplAverageRelativeError_32u_C3R . . . . .  | 2111 |
| 7.131.2.32 npplAverageRelativeError_32u_C4R . . . . .  | 2111 |
| 7.131.2.33 npplAverageRelativeError_64f_C1R . . . . .  | 2112 |

|            |   |      |
|------------|---|------|
| 7.131.2.34 | <code>ippiAverageRelativeError_64f_C2R</code> | 2112 |
| 7.131.2.35 | <code>ippiAverageRelativeError_64f_C3R</code> | 2113 |
| 7.131.2.36 | <code>ippiAverageRelativeError_64f_C4R</code> | 2113 |
| 7.131.2.37 | <code>ippiAverageRelativeError_8s_C1R</code>  | 2114 |
| 7.131.2.38 | <code>ippiAverageRelativeError_8s_C2R</code>  | 2114 |
| 7.131.2.39 | <code>ippiAverageRelativeError_8s_C3R</code>  | 2115 |
| 7.131.2.40 | <code>ippiAverageRelativeError_8s_C4R</code>  | 2115 |
| 7.131.2.41 | <code>ippiAverageRelativeError_8u_C1R</code>  | 2116 |
| 7.131.2.42 | <code>ippiAverageRelativeError_8u_C2R</code>  | 2116 |
| 7.131.2.43 | <code>ippiAverageRelativeError_8u_C3R</code>  | 2116 |
| 7.131.2.44 | <code>ippiAverageRelativeError_8u_C4R</code>  | 2117 |
| 7.132      | Memory Management                             | 2118 |
| 7.132.1    | Detailed Description                          | 2120 |
| 7.132.2    | Function Documentation                        | 2120 |
| 7.132.2.1  | <code>ippiFree</code>                         | 2120 |
| 7.132.2.2  | <code>ippiMalloc_16s_C1</code>                | 2120 |
| 7.132.2.3  | <code>ippiMalloc_16s_C2</code>                | 2121 |
| 7.132.2.4  | <code>ippiMalloc_16s_C4</code>                | 2121 |
| 7.132.2.5  | <code>ippiMalloc_16sc_C1</code>               | 2121 |
| 7.132.2.6  | <code>ippiMalloc_16sc_C2</code>               | 2121 |
| 7.132.2.7  | <code>ippiMalloc_16sc_C3</code>               | 2122 |
| 7.132.2.8  | <code>ippiMalloc_16sc_C4</code>               | 2122 |
| 7.132.2.9  | <code>ippiMalloc_16u_C1</code>                | 2122 |
| 7.132.2.10 | <code>ippiMalloc_16u_C2</code>                | 2123 |
| 7.132.2.11 | <code>ippiMalloc_16u_C3</code>                | 2123 |
| 7.132.2.12 | <code>ippiMalloc_16u_C4</code>                | 2123 |
| 7.132.2.13 | <code>ippiMalloc_32f_C1</code>                | 2123 |
| 7.132.2.14 | <code>ippiMalloc_32f_C2</code>                | 2124 |
| 7.132.2.15 | <code>ippiMalloc_32f_C3</code>                | 2124 |
| 7.132.2.16 | <code>ippiMalloc_32f_C4</code>                | 2124 |
| 7.132.2.17 | <code>ippiMalloc_32fc_C1</code>               | 2125 |
| 7.132.2.18 | <code>ippiMalloc_32fc_C2</code>               | 2125 |
| 7.132.2.19 | <code>ippiMalloc_32fc_C3</code>               | 2125 |
| 7.132.2.20 | <code>ippiMalloc_32fc_C4</code>               | 2125 |
| 7.132.2.21 | <code>ippiMalloc_32s_C1</code>                | 2126 |
| 7.132.2.22 | <code>ippiMalloc_32s_C3</code>                | 2126 |

|            |  |      |
|------------|--|------|
| 7.132.2.23 | npapiMalloc_32s_C4 . . . . .               | 2126 |
| 7.132.2.24 | npapiMalloc_32sc_C1 . . . . .              | 2127 |
| 7.132.2.25 | npapiMalloc_32sc_C2 . . . . .              | 2127 |
| 7.132.2.26 | npapiMalloc_32sc_C3 . . . . .              | 2127 |
| 7.132.2.27 | npapiMalloc_32sc_C4 . . . . .              | 2127 |
| 7.132.2.28 | npapiMalloc_8u_C1 . . . . .                | 2128 |
| 7.132.2.29 | npapiMalloc_8u_C2 . . . . .                | 2128 |
| 7.132.2.30 | npapiMalloc_8u_C3 . . . . .                | 2128 |
| 7.132.2.31 | npapiMalloc_8u_C4 . . . . .                | 2129 |
| 7.133      | Threshold and Compare Operations . . . . . | 2130 |
| 7.133.1    | Detailed Description . . . . .             | 2130 |
| 7.134      | Threshold Operations . . . . .             | 2131 |
| 7.134.1    | Detailed Description . . . . .             | 2145 |
| 7.134.2    | Function Documentation . . . . .           | 2145 |
| 7.134.2.1  | npapiThreshold_16s_AC4IR . . . . .         | 2145 |
| 7.134.2.2  | npapiThreshold_16s_AC4R . . . . .          | 2145 |
| 7.134.2.3  | npapiThreshold_16s_C1IR . . . . .          | 2146 |
| 7.134.2.4  | npapiThreshold_16s_C1R . . . . .           | 2146 |
| 7.134.2.5  | npapiThreshold_16s_C3IR . . . . .          | 2147 |
| 7.134.2.6  | npapiThreshold_16s_C3R . . . . .           | 2147 |
| 7.134.2.7  | npapiThreshold_16u_AC4IR . . . . .         | 2148 |
| 7.134.2.8  | npapiThreshold_16u_AC4R . . . . .          | 2148 |
| 7.134.2.9  | npapiThreshold_16u_C1IR . . . . .          | 2149 |
| 7.134.2.10 | npapiThreshold_16u_C1R . . . . .           | 2149 |
| 7.134.2.11 | npapiThreshold_16u_C3IR . . . . .          | 2149 |
| 7.134.2.12 | npapiThreshold_16u_C3R . . . . .           | 2150 |
| 7.134.2.13 | npapiThreshold_32f_AC4IR . . . . .         | 2150 |
| 7.134.2.14 | npapiThreshold_32f_AC4R . . . . .          | 2151 |
| 7.134.2.15 | npapiThreshold_32f_C1IR . . . . .          | 2151 |
| 7.134.2.16 | npapiThreshold_32f_C1R . . . . .           | 2152 |
| 7.134.2.17 | npapiThreshold_32f_C3IR . . . . .          | 2152 |
| 7.134.2.18 | npapiThreshold_32f_C3R . . . . .           | 2153 |
| 7.134.2.19 | npapiThreshold_8u_AC4IR . . . . .          | 2153 |
| 7.134.2.20 | npapiThreshold_8u_AC4R . . . . .           | 2154 |
| 7.134.2.21 | npapiThreshold_8u_C1IR . . . . .           | 2154 |
| 7.134.2.22 | npapiThreshold_8u_C1R . . . . .            | 2155 |

|            |                                |      |
|------------|--------------------------------|------|
| 7.134.2.23 | nppiThreshold_8u_C3IR          | 2155 |
| 7.134.2.24 | nppiThreshold_8u_C3R           | 2156 |
| 7.134.2.25 | nppiThreshold_GT_16s_AC4IR     | 2156 |
| 7.134.2.26 | nppiThreshold_GT_16s_AC4R      | 2156 |
| 7.134.2.27 | nppiThreshold_GT_16s_C1IR      | 2157 |
| 7.134.2.28 | nppiThreshold_GT_16s_C1R       | 2157 |
| 7.134.2.29 | nppiThreshold_GT_16s_C3IR      | 2158 |
| 7.134.2.30 | nppiThreshold_GT_16s_C3R       | 2158 |
| 7.134.2.31 | nppiThreshold_GT_16u_AC4IR     | 2158 |
| 7.134.2.32 | nppiThreshold_GT_16u_AC4R      | 2159 |
| 7.134.2.33 | nppiThreshold_GT_16u_C1IR      | 2159 |
| 7.134.2.34 | nppiThreshold_GT_16u_C1R       | 2160 |
| 7.134.2.35 | nppiThreshold_GT_16u_C3IR      | 2160 |
| 7.134.2.36 | nppiThreshold_GT_16u_C3R       | 2160 |
| 7.134.2.37 | nppiThreshold_GT_32f_AC4IR     | 2161 |
| 7.134.2.38 | nppiThreshold_GT_32f_AC4R      | 2161 |
| 7.134.2.39 | nppiThreshold_GT_32f_C1IR      | 2162 |
| 7.134.2.40 | nppiThreshold_GT_32f_C1R       | 2162 |
| 7.134.2.41 | nppiThreshold_GT_32f_C3IR      | 2162 |
| 7.134.2.42 | nppiThreshold_GT_32f_C3R       | 2163 |
| 7.134.2.43 | nppiThreshold_GT_8u_AC4IR      | 2163 |
| 7.134.2.44 | nppiThreshold_GT_8u_AC4R       | 2164 |
| 7.134.2.45 | nppiThreshold_GT_8u_C1IR       | 2164 |
| 7.134.2.46 | nppiThreshold_GT_8u_C1R        | 2164 |
| 7.134.2.47 | nppiThreshold_GT_8u_C3IR       | 2165 |
| 7.134.2.48 | nppiThreshold_GT_8u_C3R        | 2165 |
| 7.134.2.49 | nppiThreshold_GTVVal_16s_AC4IR | 2166 |
| 7.134.2.50 | nppiThreshold_GTVVal_16s_AC4R  | 2166 |
| 7.134.2.51 | nppiThreshold_GTVVal_16s_C1IR  | 2166 |
| 7.134.2.52 | nppiThreshold_GTVVal_16s_C1R   | 2167 |
| 7.134.2.53 | nppiThreshold_GTVVal_16s_C3IR  | 2167 |
| 7.134.2.54 | nppiThreshold_GTVVal_16s_C3R   | 2168 |
| 7.134.2.55 | nppiThreshold_GTVVal_16u_AC4IR | 2168 |
| 7.134.2.56 | nppiThreshold_GTVVal_16u_AC4R  | 2168 |
| 7.134.2.57 | nppiThreshold_GTVVal_16u_C1IR  | 2169 |
| 7.134.2.58 | nppiThreshold_GTVVal_16u_C1R   | 2169 |



|   |      |
|---|------|
| 7.134.2.59nppiThreshold_GTVal_16u_C3IR . . . . .  | 2170 |
| 7.134.2.60nppiThreshold_GTVal_16u_C3R . . . . .   | 2170 |
| 7.134.2.61nppiThreshold_GTVal_32f_AC4IR . . . . . | 2171 |
| 7.134.2.62nppiThreshold_GTVal_32f_AC4R . . . . .  | 2171 |
| 7.134.2.63nppiThreshold_GTVal_32f_C1IR . . . . .  | 2171 |
| 7.134.2.64nppiThreshold_GTVal_32f_C1R . . . . .   | 2172 |
| 7.134.2.65nppiThreshold_GTVal_32f_C3IR . . . . .  | 2172 |
| 7.134.2.66nppiThreshold_GTVal_32f_C3R . . . . .   | 2173 |
| 7.134.2.67nppiThreshold_GTVal_8u_AC4IR . . . . .  | 2173 |
| 7.134.2.68nppiThreshold_GTVal_8u_AC4R . . . . .   | 2173 |
| 7.134.2.69nppiThreshold_GTVal_8u_C1IR . . . . .   | 2174 |
| 7.134.2.70nppiThreshold_GTVal_8u_C1R . . . . .    | 2174 |
| 7.134.2.71nppiThreshold_GTVal_8u_C3IR . . . . .   | 2175 |
| 7.134.2.72nppiThreshold_GTVal_8u_C3R . . . . .    | 2175 |
| 7.134.2.73nppiThreshold_LT_16s_AC4IR . . . . .    | 2176 |
| 7.134.2.74nppiThreshold_LT_16s_AC4R . . . . .     | 2176 |
| 7.134.2.75nppiThreshold_LT_16s_C1IR . . . . .     | 2176 |
| 7.134.2.76nppiThreshold_LT_16s_C1R . . . . .      | 2177 |
| 7.134.2.77nppiThreshold_LT_16s_C3IR . . . . .     | 2177 |
| 7.134.2.78nppiThreshold_LT_16s_C3R . . . . .      | 2178 |
| 7.134.2.79nppiThreshold_LT_16u_AC4IR . . . . .    | 2178 |
| 7.134.2.80nppiThreshold_LT_16u_AC4R . . . . .     | 2178 |
| 7.134.2.81nppiThreshold_LT_16u_C1IR . . . . .     | 2179 |
| 7.134.2.82nppiThreshold_LT_16u_C1R . . . . .      | 2179 |
| 7.134.2.83nppiThreshold_LT_16u_C3IR . . . . .     | 2180 |
| 7.134.2.84nppiThreshold_LT_16u_C3R . . . . .      | 2180 |
| 7.134.2.85nppiThreshold_LT_32f_AC4IR . . . . .    | 2180 |
| 7.134.2.86nppiThreshold_LT_32f_AC4R . . . . .     | 2181 |
| 7.134.2.87nppiThreshold_LT_32f_C1IR . . . . .     | 2181 |
| 7.134.2.88nppiThreshold_LT_32f_C1R . . . . .      | 2182 |
| 7.134.2.89nppiThreshold_LT_32f_C3IR . . . . .     | 2182 |
| 7.134.2.90nppiThreshold_LT_32f_C3R . . . . .      | 2182 |
| 7.134.2.91nppiThreshold_LT_8u_AC4IR . . . . .     | 2183 |
| 7.134.2.92nppiThreshold_LT_8u_AC4R . . . . .      | 2183 |
| 7.134.2.93nppiThreshold_LT_8u_C1IR . . . . .      | 2184 |
| 7.134.2.94nppiThreshold_LT_8u_C1R . . . . .       | 2184 |

|             |                                   |      |
|-------------|-----------------------------------|------|
| 7.134.2.95  | ppiThreshold_LT_8u_C3IR           | 2184 |
| 7.134.2.96  | ppiThreshold_LT_8u_C3R            | 2185 |
| 7.134.2.97  | ppiThreshold_LTVa1_16s_AC4IR      | 2185 |
| 7.134.2.98  | ppiThreshold_LTVa1_16s_AC4R       | 2186 |
| 7.134.2.99  | ppiThreshold_LTVa1_16s_C1IR       | 2186 |
| 7.134.2.100 | ppiThreshold_LTVa1_16s_C1R        | 2186 |
| 7.134.2.101 | ppiThreshold_LTVa1_16s_C3IR       | 2187 |
| 7.134.2.102 | ppiThreshold_LTVa1_16s_C3R        | 2187 |
| 7.134.2.103 | ppiThreshold_LTVa1_16u_AC4IR      | 2188 |
| 7.134.2.104 | ppiThreshold_LTVa1_16u_AC4R       | 2188 |
| 7.134.2.105 | ppiThreshold_LTVa1_16u_C1IR       | 2189 |
| 7.134.2.106 | ppiThreshold_LTVa1_16u_C1R        | 2189 |
| 7.134.2.107 | ppiThreshold_LTVa1_16u_C3IR       | 2189 |
| 7.134.2.108 | ppiThreshold_LTVa1_16u_C3R        | 2190 |
| 7.134.2.109 | ppiThreshold_LTVa1_32f_AC4IR      | 2190 |
| 7.134.2.110 | ppiThreshold_LTVa1_32f_AC4R       | 2191 |
| 7.134.2.111 | ppiThreshold_LTVa1_32f_C1IR       | 2191 |
| 7.134.2.112 | ppiThreshold_LTVa1_32f_C1R        | 2191 |
| 7.134.2.113 | ppiThreshold_LTVa1_32f_C3IR       | 2192 |
| 7.134.2.114 | ppiThreshold_LTVa1_32f_C3R        | 2192 |
| 7.134.2.115 | ppiThreshold_LTVa1_8u_AC4IR       | 2193 |
| 7.134.2.116 | ppiThreshold_LTVa1_8u_AC4R        | 2193 |
| 7.134.2.117 | ppiThreshold_LTVa1_8u_C1IR        | 2194 |
| 7.134.2.118 | ppiThreshold_LTVa1_8u_C1R         | 2194 |
| 7.134.2.119 | ppiThreshold_LTVa1_8u_C3IR        | 2194 |
| 7.134.2.120 | ppiThreshold_LTVa1_8u_C3R         | 2195 |
| 7.134.2.121 | ppiThreshold_LTVa1GTVal_16s_AC4IR | 2195 |
| 7.134.2.122 | ppiThreshold_LTVa1GTVal_16s_AC4R  | 2196 |
| 7.134.2.123 | ppiThreshold_LTVa1GTVal_16s_C1IR  | 2196 |
| 7.134.2.124 | ppiThreshold_LTVa1GTVal_16s_C1R   | 2197 |
| 7.134.2.125 | ppiThreshold_LTVa1GTVal_16s_C3IR  | 2197 |
| 7.134.2.126 | ppiThreshold_LTVa1GTVal_16s_C3R   | 2198 |
| 7.134.2.127 | ppiThreshold_LTVa1GTVal_16u_AC4IR | 2198 |
| 7.134.2.128 | ppiThreshold_LTVa1GTVal_16u_AC4R  | 2199 |
| 7.134.2.129 | ppiThreshold_LTVa1GTVal_16u_C1IR  | 2199 |
| 7.134.2.130 | ppiThreshold_LTVa1GTVal_16u_C1R   | 2200 |

|   |      |
|---|------|
| 7.134.2.131ppiThreshold_LTVaGTVaL_16u_C3IR . . . . .  | 2200 |
| 7.134.2.132ppiThreshold_LTVaGTVaL_16u_C3R . . . . .   | 2201 |
| 7.134.2.133ppiThreshold_LTVaGTVaL_32f_AC4IR . . . . . | 2201 |
| 7.134.2.134ppiThreshold_LTVaGTVaL_32f_AC4R . . . . .  | 2202 |
| 7.134.2.135ppiThreshold_LTVaGTVaL_32f_C1IR . . . . .  | 2202 |
| 7.134.2.136ppiThreshold_LTVaGTVaL_32f_C1R . . . . .   | 2203 |
| 7.134.2.137ppiThreshold_LTVaGTVaL_32f_C3IR . . . . .  | 2203 |
| 7.134.2.138ppiThreshold_LTVaGTVaL_32f_C3R . . . . .   | 2204 |
| 7.134.2.139ppiThreshold_LTVaGTVaL_8u_AC4IR . . . . .  | 2204 |
| 7.134.2.140ppiThreshold_LTVaGTVaL_8u_AC4R . . . . .   | 2205 |
| 7.134.2.141ppiThreshold_LTVaGTVaL_8u_C1IR . . . . .   | 2205 |
| 7.134.2.142ppiThreshold_LTVaGTVaL_8u_C1R . . . . .    | 2206 |
| 7.134.2.143ppiThreshold_LTVaGTVaL_8u_C3IR . . . . .   | 2206 |
| 7.134.2.144ppiThreshold_LTVaGTVaL_8u_C3R . . . . .    | 2207 |
| 7.134.2.145ppiThreshold_Val_16s_AC4IR . . . . .       | 2207 |
| 7.134.2.146ppiThreshold_Val_16s_AC4R . . . . .        | 2208 |
| 7.134.2.147ppiThreshold_Val_16s_C1IR . . . . .        | 2208 |
| 7.134.2.148ppiThreshold_Val_16s_C1R . . . . .         | 2209 |
| 7.134.2.149ppiThreshold_Val_16s_C3IR . . . . .        | 2209 |
| 7.134.2.150ppiThreshold_Val_16s_C3R . . . . .         | 2210 |
| 7.134.2.151ppiThreshold_Val_16u_AC4IR . . . . .       | 2210 |
| 7.134.2.152ppiThreshold_Val_16u_AC4R . . . . .        | 2211 |
| 7.134.2.153ppiThreshold_Val_16u_C1IR . . . . .        | 2211 |
| 7.134.2.154ppiThreshold_Val_16u_C1R . . . . .         | 2212 |
| 7.134.2.155ppiThreshold_Val_16u_C3IR . . . . .        | 2212 |
| 7.134.2.156ppiThreshold_Val_16u_C3R . . . . .         | 2213 |
| 7.134.2.157ppiThreshold_Val_32f_AC4IR . . . . .       | 2213 |
| 7.134.2.158ppiThreshold_Val_32f_AC4R . . . . .        | 2214 |
| 7.134.2.159ppiThreshold_Val_32f_C1IR . . . . .        | 2214 |
| 7.134.2.160ppiThreshold_Val_32f_C1R . . . . .         | 2215 |
| 7.134.2.161ppiThreshold_Val_32f_C3IR . . . . .        | 2215 |
| 7.134.2.162ppiThreshold_Val_32f_C3R . . . . .         | 2216 |
| 7.134.2.163ppiThreshold_Val_8u_AC4IR . . . . .        | 2216 |
| 7.134.2.164ppiThreshold_Val_8u_AC4R . . . . .         | 2217 |
| 7.134.2.165ppiThreshold_Val_8u_C1IR . . . . .         | 2217 |
| 7.134.2.166ppiThreshold_Val_8u_C1R . . . . .          | 2218 |

|   |      |
|---|------|
| 7.134.2.167 nppiThreshold_Val_8u_C3IR . . . . . | 2218 |
| 7.134.2.168 nppiThreshold_Val_8u_C3R . . . . .  | 2219 |
| 7.135 Compare Operations . . . . .              | 2220 |
| 7.135.1 Detailed Description . . . . .          | 2223 |
| 7.135.2 Function Documentation . . . . .        | 2223 |
| 7.135.2.1 nppiCompare_16s_AC4R . . . . .        | 2223 |
| 7.135.2.2 nppiCompare_16s_C1R . . . . .         | 2224 |
| 7.135.2.3 nppiCompare_16s_C3R . . . . .         | 2224 |
| 7.135.2.4 nppiCompare_16s_C4R . . . . .         | 2225 |
| 7.135.2.5 nppiCompare_16u_AC4R . . . . .        | 2225 |
| 7.135.2.6 nppiCompare_16u_C1R . . . . .         | 2226 |
| 7.135.2.7 nppiCompare_16u_C3R . . . . .         | 2226 |
| 7.135.2.8 nppiCompare_16u_C4R . . . . .         | 2227 |
| 7.135.2.9 nppiCompare_32f_AC4R . . . . .        | 2227 |
| 7.135.2.10 nppiCompare_32f_C1R . . . . .        | 2228 |
| 7.135.2.11 nppiCompare_32f_C3R . . . . .        | 2228 |
| 7.135.2.12 nppiCompare_32f_C4R . . . . .        | 2229 |
| 7.135.2.13 nppiCompare_8u_AC4R . . . . .        | 2229 |
| 7.135.2.14 nppiCompare_8u_C1R . . . . .         | 2230 |
| 7.135.2.15 nppiCompare_8u_C3R . . . . .         | 2230 |
| 7.135.2.16 nppiCompare_8u_C4R . . . . .         | 2231 |
| 7.135.2.17 nppiCompareC_16s_AC4R . . . . .      | 2231 |
| 7.135.2.18 nppiCompareC_16s_C1R . . . . .       | 2231 |
| 7.135.2.19 nppiCompareC_16s_C3R . . . . .       | 2232 |
| 7.135.2.20 nppiCompareC_16s_C4R . . . . .       | 2232 |
| 7.135.2.21 nppiCompareC_16u_AC4R . . . . .      | 2233 |
| 7.135.2.22 nppiCompareC_16u_C1R . . . . .       | 2233 |
| 7.135.2.23 nppiCompareC_16u_C3R . . . . .       | 2234 |
| 7.135.2.24 nppiCompareC_16u_C4R . . . . .       | 2234 |
| 7.135.2.25 nppiCompareC_32f_AC4R . . . . .      | 2234 |
| 7.135.2.26 nppiCompareC_32f_C1R . . . . .       | 2235 |
| 7.135.2.27 nppiCompareC_32f_C3R . . . . .       | 2235 |
| 7.135.2.28 nppiCompareC_32f_C4R . . . . .       | 2236 |
| 7.135.2.29 nppiCompareC_8u_AC4R . . . . .       | 2236 |
| 7.135.2.30 nppiCompareC_8u_C1R . . . . .        | 2237 |
| 7.135.2.31 nppiCompareC_8u_C3R . . . . .        | 2237 |

|            |   |      |
|------------|---|------|
| 7.135.2.32 | <a href="#">ippiCompareC_8u_C4R</a>               | 2237 |
| 7.135.2.33 | <a href="#">ippiCompareEqualEps_32f_AC4R</a>      | 2238 |
| 7.135.2.34 | <a href="#">ippiCompareEqualEps_32f_C1R</a>       | 2238 |
| 7.135.2.35 | <a href="#">ippiCompareEqualEps_32f_C3R</a>       | 2239 |
| 7.135.2.36 | <a href="#">ippiCompareEqualEps_32f_C4R</a>       | 2239 |
| 7.135.2.37 | <a href="#">ippiCompareEqualEpsC_32f_AC4R</a>     | 2240 |
| 7.135.2.38 | <a href="#">ippiCompareEqualEpsC_32f_C1R</a>      | 2240 |
| 7.135.2.39 | <a href="#">ippiCompareEqualEpsC_32f_C3R</a>      | 2241 |
| 7.135.2.40 | <a href="#">ippiCompareEqualEpsC_32f_C4R</a>      | 2241 |
| 7.136      | <a href="#">NPP Signal Processing</a>             | 2243 |
| 7.137      | <a href="#">Arithmetic and Logical Operations</a> | 2244 |
| 7.138      | <a href="#">Arithmetic Operations</a>             | 2245 |
| 7.139      | <a href="#">AddC</a>                              | 2247 |
| 7.139.1    | <a href="#">Detailed Description</a>              | 2248 |
| 7.139.2    | <a href="#">Function Documentation</a>            | 2248 |
| 7.139.2.1  | <a href="#">nppsAddC_16s_ISfs</a>                 | 2248 |
| 7.139.2.2  | <a href="#">nppsAddC_16s_Sfs</a>                  | 2249 |
| 7.139.2.3  | <a href="#">nppsAddC_16sc_ISfs</a>                | 2249 |
| 7.139.2.4  | <a href="#">nppsAddC_16sc_Sfs</a>                 | 2249 |
| 7.139.2.5  | <a href="#">nppsAddC_16u_ISfs</a>                 | 2250 |
| 7.139.2.6  | <a href="#">nppsAddC_16u_Sfs</a>                  | 2250 |
| 7.139.2.7  | <a href="#">nppsAddC_32f</a>                      | 2250 |
| 7.139.2.8  | <a href="#">nppsAddC_32f_I</a>                    | 2251 |
| 7.139.2.9  | <a href="#">nppsAddC_32fc</a>                     | 2251 |
| 7.139.2.10 | <a href="#">nppsAddC_32fc_I</a>                   | 2251 |
| 7.139.2.11 | <a href="#">nppsAddC_32s_ISfs</a>                 | 2251 |
| 7.139.2.12 | <a href="#">nppsAddC_32s_Sfs</a>                  | 2252 |
| 7.139.2.13 | <a href="#">nppsAddC_32sc_ISfs</a>                | 2252 |
| 7.139.2.14 | <a href="#">nppsAddC_32sc_Sfs</a>                 | 2253 |
| 7.139.2.15 | <a href="#">nppsAddC_64f</a>                      | 2253 |
| 7.139.2.16 | <a href="#">nppsAddC_64f_I</a>                    | 2253 |
| 7.139.2.17 | <a href="#">nppsAddC_64fc</a>                     | 2254 |
| 7.139.2.18 | <a href="#">nppsAddC_64fc_I</a>                   | 2254 |
| 7.139.2.19 | <a href="#">nppsAddC_8u_ISfs</a>                  | 2254 |
| 7.139.2.20 | <a href="#">nppsAddC_8u_Sfs</a>                   | 2255 |
| 7.140      | <a href="#">AddProductC</a>                       | 2256 |

|   |      |
|---|------|
| 7.140.1 Detailed Description . . . . .    | 2256 |
| 7.140.2 Function Documentation . . . . .  | 2256 |
| 7.140.2.1 nppsAddProductC_32f . . . . .   | 2256 |
| 7.141 MulC . . . . .                      | 2257 |
| 7.141.1 Detailed Description . . . . .    | 2258 |
| 7.141.2 Function Documentation . . . . .  | 2258 |
| 7.141.2.1 nppsMulC_16s_ISfs . . . . .     | 2258 |
| 7.141.2.2 nppsMulC_16s_Sfs . . . . .      | 2259 |
| 7.141.2.3 nppsMulC_16sc_ISfs . . . . .    | 2259 |
| 7.141.2.4 nppsMulC_16sc_Sfs . . . . .     | 2260 |
| 7.141.2.5 nppsMulC_16u_ISfs . . . . .     | 2260 |
| 7.141.2.6 nppsMulC_16u_Sfs . . . . .      | 2260 |
| 7.141.2.7 nppsMulC_32f . . . . .          | 2261 |
| 7.141.2.8 nppsMulC_32f16s_Sfs . . . . .   | 2261 |
| 7.141.2.9 nppsMulC_32f_I . . . . .        | 2261 |
| 7.141.2.10 nppsMulC_32fc . . . . .        | 2262 |
| 7.141.2.11 nppsMulC_32fc_I . . . . .      | 2262 |
| 7.141.2.12 nppsMulC_32s_ISfs . . . . .    | 2262 |
| 7.141.2.13 nppsMulC_32s_Sfs . . . . .     | 2263 |
| 7.141.2.14 nppsMulC_32sc_ISfs . . . . .   | 2263 |
| 7.141.2.15 nppsMulC_32sc_Sfs . . . . .    | 2263 |
| 7.141.2.16 nppsMulC_64f . . . . .         | 2264 |
| 7.141.2.17 nppsMulC_64f64s_ISfs . . . . . | 2264 |
| 7.141.2.18 nppsMulC_64f_I . . . . .       | 2264 |
| 7.141.2.19 nppsMulC_64fc . . . . .        | 2265 |
| 7.141.2.20 nppsMulC_64fc_I . . . . .      | 2265 |
| 7.141.2.21 nppsMulC_8u_ISfs . . . . .     | 2265 |
| 7.141.2.22 nppsMulC_8u_Sfs . . . . .      | 2266 |
| 7.141.2.23 nppsMulC_Low_32f16s . . . . .  | 2266 |
| 7.142 SubC . . . . .                      | 2267 |
| 7.142.1 Detailed Description . . . . .    | 2268 |
| 7.142.2 Function Documentation . . . . .  | 2268 |
| 7.142.2.1 nppsSubC_16s_ISfs . . . . .     | 2268 |
| 7.142.2.2 nppsSubC_16s_Sfs . . . . .      | 2269 |
| 7.142.2.3 nppsSubC_16sc_ISfs . . . . .    | 2269 |
| 7.142.2.4 nppsSubC_16sc_Sfs . . . . .     | 2269 |

|  |      |
|--|------|
| 7.142.2.5 nppsSubC_16u_ISfs . . . . .      | 2270 |
| 7.142.2.6 nppsSubC_16u_Sfs . . . . .       | 2270 |
| 7.142.2.7 nppsSubC_32f . . . . .           | 2270 |
| 7.142.2.8 nppsSubC_32f_I . . . . .         | 2271 |
| 7.142.2.9 nppsSubC_32fc . . . . .          | 2271 |
| 7.142.2.10 nppsSubC_32fc_I . . . . .       | 2271 |
| 7.142.2.11 nppsSubC_32s_ISfs . . . . .     | 2271 |
| 7.142.2.12 nppsSubC_32s_Sfs . . . . .      | 2272 |
| 7.142.2.13 nppsSubC_32sc_ISfs . . . . .    | 2272 |
| 7.142.2.14 nppsSubC_32sc_Sfs . . . . .     | 2273 |
| 7.142.2.15 nppsSubC_64f . . . . .          | 2273 |
| 7.142.2.16 nppsSubC_64f_I . . . . .        | 2273 |
| 7.142.2.17 nppsSubC_64fc . . . . .         | 2274 |
| 7.142.2.18 nppsSubC_64fc_I . . . . .       | 2274 |
| 7.142.2.19 nppsSubC_8u_ISfs . . . . .      | 2274 |
| 7.142.2.20 nppsSubC_8u_Sfs . . . . .       | 2275 |
| 7.143 SubCRev . . . . .                    | 2276 |
| 7.143.1 Detailed Description . . . . .     | 2277 |
| 7.143.2 Function Documentation . . . . .   | 2277 |
| 7.143.2.1 nppsSubCRev_16s_ISfs . . . . .   | 2277 |
| 7.143.2.2 nppsSubCRev_16s_Sfs . . . . .    | 2278 |
| 7.143.2.3 nppsSubCRev_16sc_ISfs . . . . .  | 2278 |
| 7.143.2.4 nppsSubCRev_16sc_Sfs . . . . .   | 2278 |
| 7.143.2.5 nppsSubCRev_16u_ISfs . . . . .   | 2279 |
| 7.143.2.6 nppsSubCRev_16u_Sfs . . . . .    | 2279 |
| 7.143.2.7 nppsSubCRev_32f . . . . .        | 2279 |
| 7.143.2.8 nppsSubCRev_32f_I . . . . .      | 2280 |
| 7.143.2.9 nppsSubCRev_32fc . . . . .       | 2280 |
| 7.143.2.10 nppsSubCRev_32fc_I . . . . .    | 2280 |
| 7.143.2.11 nppsSubCRev_32s_ISfs . . . . .  | 2281 |
| 7.143.2.12 nppsSubCRev_32s_Sfs . . . . .   | 2281 |
| 7.143.2.13 nppsSubCRev_32sc_ISfs . . . . . | 2281 |
| 7.143.2.14 nppsSubCRev_32sc_Sfs . . . . .  | 2282 |
| 7.143.2.15 nppsSubCRev_64f . . . . .       | 2282 |
| 7.143.2.16 nppsSubCRev_64f_I . . . . .     | 2282 |
| 7.143.2.17 nppsSubCRev_64fc . . . . .      | 2283 |

|            |                                  |      |
|------------|----------------------------------|------|
| 7.143.2.18 | nppsSubCRev_64fc_I . . . . .     | 2283 |
| 7.143.2.19 | nppsSubCRev_8u_ISfs . . . . .    | 2283 |
| 7.143.2.20 | nppsSubCRev_8u_Sfs . . . . .     | 2284 |
| 7.144      | DivC . . . . .                   | 2285 |
| 7.144.1    | Detailed Description . . . . .   | 2286 |
| 7.144.2    | Function Documentation . . . . . | 2286 |
| 7.144.2.1  | nppsDivC_16s_ISfs . . . . .      | 2286 |
| 7.144.2.2  | nppsDivC_16s_Sfs . . . . .       | 2286 |
| 7.144.2.3  | nppsDivC_16sc_ISfs . . . . .     | 2287 |
| 7.144.2.4  | nppsDivC_16sc_Sfs . . . . .      | 2287 |
| 7.144.2.5  | nppsDivC_16u_ISfs . . . . .      | 2287 |
| 7.144.2.6  | nppsDivC_16u_Sfs . . . . .       | 2288 |
| 7.144.2.7  | nppsDivC_32f . . . . .           | 2288 |
| 7.144.2.8  | nppsDivC_32f_I . . . . .         | 2288 |
| 7.144.2.9  | nppsDivC_32fc . . . . .          | 2289 |
| 7.144.2.10 | nppsDivC_32fc_I . . . . .        | 2289 |
| 7.144.2.11 | nppsDivC_64f . . . . .           | 2289 |
| 7.144.2.12 | nppsDivC_64f_I . . . . .         | 2290 |
| 7.144.2.13 | nppsDivC_64fc . . . . .          | 2290 |
| 7.144.2.14 | nppsDivC_64fc_I . . . . .        | 2290 |
| 7.144.2.15 | nppsDivC_8u_ISfs . . . . .       | 2290 |
| 7.144.2.16 | nppsDivC_8u_Sfs . . . . .        | 2291 |
| 7.145      | DivCRev . . . . .                | 2292 |
| 7.145.1    | Detailed Description . . . . .   | 2292 |
| 7.145.2    | Function Documentation . . . . . | 2292 |
| 7.145.2.1  | nppsDivCRev_16u . . . . .        | 2292 |
| 7.145.2.2  | nppsDivCRev_16u_I . . . . .      | 2292 |
| 7.145.2.3  | nppsDivCRev_32f . . . . .        | 2293 |
| 7.145.2.4  | nppsDivCRev_32f_I . . . . .      | 2293 |
| 7.146      | Add . . . . .                    | 2294 |
| 7.146.1    | Detailed Description . . . . .   | 2296 |
| 7.146.2    | Function Documentation . . . . . | 2296 |
| 7.146.2.1  | nppsAdd_16s . . . . .            | 2296 |
| 7.146.2.2  | nppsAdd_16s32f . . . . .         | 2296 |
| 7.146.2.3  | nppsAdd_16s32s_I . . . . .       | 2297 |
| 7.146.2.4  | nppsAdd_16s_I . . . . .          | 2297 |



|   |      |
|---|------|
| 7.146.2.5 nppsAdd_16s_ISfs . . . . .          | 2297 |
| 7.146.2.6 nppsAdd_16s_Sfs . . . . .           | 2298 |
| 7.146.2.7 nppsAdd_16sc_ISfs . . . . .         | 2298 |
| 7.146.2.8 nppsAdd_16sc_Sfs . . . . .          | 2298 |
| 7.146.2.9 nppsAdd_16u . . . . .               | 2299 |
| 7.146.2.10 nppsAdd_16u_ISfs . . . . .         | 2299 |
| 7.146.2.11 nppsAdd_16u_Sfs . . . . .          | 2299 |
| 7.146.2.12 nppsAdd_32f . . . . .              | 2300 |
| 7.146.2.13 nppsAdd_32f_I . . . . .            | 2300 |
| 7.146.2.14 nppsAdd_32fc . . . . .             | 2300 |
| 7.146.2.15 nppsAdd_32fc_I . . . . .           | 2301 |
| 7.146.2.16 nppsAdd_32s_ISfs . . . . .         | 2301 |
| 7.146.2.17 nppsAdd_32s_Sfs . . . . .          | 2301 |
| 7.146.2.18 nppsAdd_32sc_ISfs . . . . .        | 2302 |
| 7.146.2.19 nppsAdd_32sc_Sfs . . . . .         | 2302 |
| 7.146.2.20 nppsAdd_32u . . . . .              | 2302 |
| 7.146.2.21 nppsAdd_64f . . . . .              | 2303 |
| 7.146.2.22 nppsAdd_64f_I . . . . .            | 2303 |
| 7.146.2.23 nppsAdd_64fc . . . . .             | 2303 |
| 7.146.2.24 nppsAdd_64fc_I . . . . .           | 2304 |
| 7.146.2.25 nppsAdd_64s_Sfs . . . . .          | 2304 |
| 7.146.2.26 nppsAdd_8u16u . . . . .            | 2304 |
| 7.146.2.27 nppsAdd_8u_ISfs . . . . .          | 2305 |
| 7.146.2.28 nppsAdd_8u_Sfs . . . . .           | 2305 |
| 7.147 AddProduct . . . . .                    | 2306 |
| 7.147.1 Detailed Description . . . . .        | 2306 |
| 7.147.2 Function Documentation . . . . .      | 2307 |
| 7.147.2.1 nppsAddProduct_16s32s_Sfs . . . . . | 2307 |
| 7.147.2.2 nppsAddProduct_16s_Sfs . . . . .    | 2307 |
| 7.147.2.3 nppsAddProduct_32f . . . . .        | 2307 |
| 7.147.2.4 nppsAddProduct_32fc . . . . .       | 2308 |
| 7.147.2.5 nppsAddProduct_32s_Sfs . . . . .    | 2308 |
| 7.147.2.6 nppsAddProduct_64f . . . . .        | 2309 |
| 7.147.2.7 nppsAddProduct_64fc . . . . .       | 2309 |
| 7.148 Mul . . . . .                           | 2310 |
| 7.148.1 Detailed Description . . . . .        | 2312 |

|   |      |
|---|------|
| 7.148.2 Function Documentation . . . . .  | 2312 |
| 7.148.2.1 nppsMul_16s . . . . .           | 2312 |
| 7.148.2.2 nppsMul_16s32f . . . . .        | 2313 |
| 7.148.2.3 nppsMul_16s32s_Sfs . . . . .    | 2313 |
| 7.148.2.4 nppsMul_16s_I . . . . .         | 2313 |
| 7.148.2.5 nppsMul_16s_ISfs . . . . .      | 2314 |
| 7.148.2.6 nppsMul_16s_Sfs . . . . .       | 2314 |
| 7.148.2.7 nppsMul_16sc_ISfs . . . . .     | 2314 |
| 7.148.2.8 nppsMul_16sc_Sfs . . . . .      | 2315 |
| 7.148.2.9 nppsMul_16u16s_Sfs . . . . .    | 2315 |
| 7.148.2.10 nppsMul_16u_ISfs . . . . .     | 2315 |
| 7.148.2.11 nppsMul_16u_Sfs . . . . .      | 2316 |
| 7.148.2.12 nppsMul_32f . . . . .          | 2316 |
| 7.148.2.13 nppsMul_32f32fc . . . . .      | 2316 |
| 7.148.2.14 nppsMul_32f32fc_I . . . . .    | 2317 |
| 7.148.2.15 nppsMul_32f_I . . . . .        | 2317 |
| 7.148.2.16 nppsMul_32fc . . . . .         | 2317 |
| 7.148.2.17 nppsMul_32fc_I . . . . .       | 2318 |
| 7.148.2.18 nppsMul_32s32sc_ISfs . . . . . | 2318 |
| 7.148.2.19 nppsMul_32s32sc_Sfs . . . . .  | 2318 |
| 7.148.2.20 nppsMul_32s_ISfs . . . . .     | 2319 |
| 7.148.2.21 nppsMul_32s_Sfs . . . . .      | 2319 |
| 7.148.2.22 nppsMul_32sc_ISfs . . . . .    | 2319 |
| 7.148.2.23 nppsMul_32sc_Sfs . . . . .     | 2320 |
| 7.148.2.24 nppsMul_64f . . . . .          | 2320 |
| 7.148.2.25 nppsMul_64f_I . . . . .        | 2320 |
| 7.148.2.26 nppsMul_64fc . . . . .         | 2321 |
| 7.148.2.27 nppsMul_64fc_I . . . . .       | 2321 |
| 7.148.2.28 nppsMul_8u16u . . . . .        | 2321 |
| 7.148.2.29 nppsMul_8u_ISfs . . . . .      | 2322 |
| 7.148.2.30 nppsMul_8u_Sfs . . . . .       | 2322 |
| 7.148.2.31 nppsMul_Low_32s_Sfs . . . . .  | 2322 |
| 7.149 Sub . . . . .                       | 2323 |
| 7.149.1 Detailed Description . . . . .    | 2324 |
| 7.149.2 Function Documentation . . . . .  | 2324 |
| 7.149.2.1 nppsSub_16s . . . . .           | 2324 |

|  |      |
|--|------|
| 7.149.2.2 nppsSub_16s32f . . . . .       | 2325 |
| 7.149.2.3 nppsSub_16s_I . . . . .        | 2325 |
| 7.149.2.4 nppsSub_16s_ISfs . . . . .     | 2325 |
| 7.149.2.5 nppsSub_16s_Sfs . . . . .      | 2326 |
| 7.149.2.6 nppsSub_16sc_ISfs . . . . .    | 2326 |
| 7.149.2.7 nppsSub_16sc_Sfs . . . . .     | 2326 |
| 7.149.2.8 nppsSub_16u_ISfs . . . . .     | 2327 |
| 7.149.2.9 nppsSub_16u_Sfs . . . . .      | 2327 |
| 7.149.2.10 nppsSub_32f . . . . .         | 2327 |
| 7.149.2.11 nppsSub_32f_I . . . . .       | 2328 |
| 7.149.2.12 nppsSub_32fc . . . . .        | 2328 |
| 7.149.2.13 nppsSub_32fc_I . . . . .      | 2328 |
| 7.149.2.14 nppsSub_32s_ISfs . . . . .    | 2329 |
| 7.149.2.15 nppsSub_32s_Sfs . . . . .     | 2329 |
| 7.149.2.16 nppsSub_32sc_ISfs . . . . .   | 2329 |
| 7.149.2.17 nppsSub_32sc_Sfs . . . . .    | 2330 |
| 7.149.2.18 nppsSub_64f . . . . .         | 2330 |
| 7.149.2.19 nppsSub_64f_I . . . . .       | 2330 |
| 7.149.2.20 nppsSub_64fc . . . . .        | 2331 |
| 7.149.2.21 nppsSub_64fc_I . . . . .      | 2331 |
| 7.149.2.22 nppsSub_8u_ISfs . . . . .     | 2331 |
| 7.149.2.23 nppsSub_8u_Sfs . . . . .      | 2332 |
| 7.150 Div . . . . .                      | 2333 |
| 7.150.1 Detailed Description . . . . .   | 2334 |
| 7.150.2 Function Documentation . . . . . | 2334 |
| 7.150.2.1 nppsDiv_16s_ISfs . . . . .     | 2334 |
| 7.150.2.2 nppsDiv_16s_Sfs . . . . .      | 2335 |
| 7.150.2.3 nppsDiv_16sc_ISfs . . . . .    | 2335 |
| 7.150.2.4 nppsDiv_16sc_Sfs . . . . .     | 2335 |
| 7.150.2.5 nppsDiv_16u_ISfs . . . . .     | 2336 |
| 7.150.2.6 nppsDiv_16u_Sfs . . . . .      | 2336 |
| 7.150.2.7 nppsDiv_32f . . . . .          | 2336 |
| 7.150.2.8 nppsDiv_32f_I . . . . .        | 2337 |
| 7.150.2.9 nppsDiv_32fc . . . . .         | 2337 |
| 7.150.2.10 nppsDiv_32fc_I . . . . .      | 2337 |
| 7.150.2.11 nppsDiv_32s16s_Sfs . . . . .  | 2337 |

|            |                                  |      |
|------------|----------------------------------|------|
| 7.150.2.12 | nppsDiv_32s_ISfs . . . . .       | 2338 |
| 7.150.2.13 | nppsDiv_32s_Sfs . . . . .        | 2338 |
| 7.150.2.14 | nppsDiv_64f . . . . .            | 2339 |
| 7.150.2.15 | nppsDiv_64f_I . . . . .          | 2339 |
| 7.150.2.16 | nppsDiv_64fc . . . . .           | 2339 |
| 7.150.2.17 | nppsDiv_64fc_I . . . . .         | 2340 |
| 7.150.2.18 | nppsDiv_8u_ISfs . . . . .        | 2340 |
| 7.150.2.19 | nppsDiv_8u_Sfs . . . . .         | 2340 |
| 7.151      | Div_Round . . . . .              | 2341 |
| 7.151.1    | Detailed Description . . . . .   | 2341 |
| 7.151.2    | Function Documentation . . . . . | 2341 |
| 7.151.2.1  | nppsDiv_Round_16s_ISfs . . . . . | 2341 |
| 7.151.2.2  | nppsDiv_Round_16s_Sfs . . . . .  | 2342 |
| 7.151.2.3  | nppsDiv_Round_16u_ISfs . . . . . | 2342 |
| 7.151.2.4  | nppsDiv_Round_16u_Sfs . . . . .  | 2342 |
| 7.151.2.5  | nppsDiv_Round_8u_ISfs . . . . .  | 2343 |
| 7.151.2.6  | nppsDiv_Round_8u_Sfs . . . . .   | 2343 |
| 7.152      | Abs . . . . .                    | 2344 |
| 7.152.1    | Detailed Description . . . . .   | 2344 |
| 7.152.2    | Function Documentation . . . . . | 2344 |
| 7.152.2.1  | nppsAbs_16s . . . . .            | 2344 |
| 7.152.2.2  | nppsAbs_16s_I . . . . .          | 2345 |
| 7.152.2.3  | nppsAbs_32f . . . . .            | 2345 |
| 7.152.2.4  | nppsAbs_32f_I . . . . .          | 2345 |
| 7.152.2.5  | nppsAbs_32s . . . . .            | 2345 |
| 7.152.2.6  | nppsAbs_32s_I . . . . .          | 2346 |
| 7.152.2.7  | nppsAbs_64f . . . . .            | 2346 |
| 7.152.2.8  | nppsAbs_64f_I . . . . .          | 2346 |
| 7.153      | Sqr . . . . .                    | 2347 |
| 7.153.1    | Detailed Description . . . . .   | 2348 |
| 7.153.2    | Function Documentation . . . . . | 2348 |
| 7.153.2.1  | nppsSqr_16s_ISfs . . . . .       | 2348 |
| 7.153.2.2  | nppsSqr_16s_Sfs . . . . .        | 2348 |
| 7.153.2.3  | nppsSqr_16sc_ISfs . . . . .      | 2348 |
| 7.153.2.4  | nppsSqr_16sc_Sfs . . . . .       | 2349 |
| 7.153.2.5  | nppsSqr_16u_ISfs . . . . .       | 2349 |

|            |                                  |      |
|------------|----------------------------------|------|
| 7.153.2.6  | nppsSqr_16u_Sfs . . . . .        | 2349 |
| 7.153.2.7  | nppsSqr_32f . . . . .            | 2350 |
| 7.153.2.8  | nppsSqr_32f_I . . . . .          | 2350 |
| 7.153.2.9  | nppsSqr_32fc . . . . .           | 2350 |
| 7.153.2.10 | nppsSqr_32fc_I . . . . .         | 2350 |
| 7.153.2.11 | nppsSqr_64f . . . . .            | 2351 |
| 7.153.2.12 | nppsSqr_64f_I . . . . .          | 2351 |
| 7.153.2.13 | nppsSqr_64fc . . . . .           | 2351 |
| 7.153.2.14 | nppsSqr_64fc_I . . . . .         | 2351 |
| 7.153.2.15 | nppsSqr_8u_ISfs . . . . .        | 2352 |
| 7.153.2.16 | nppsSqr_8u_Sfs . . . . .         | 2352 |
| 7.154      | Sqrt . . . . .                   | 2353 |
| 7.154.1    | Detailed Description . . . . .   | 2354 |
| 7.154.2    | Function Documentation . . . . . | 2354 |
| 7.154.2.1  | nppsSqrt_16s_ISfs . . . . .      | 2354 |
| 7.154.2.2  | nppsSqrt_16s_Sfs . . . . .       | 2354 |
| 7.154.2.3  | nppsSqrt_16sc_ISfs . . . . .     | 2355 |
| 7.154.2.4  | nppsSqrt_16sc_Sfs . . . . .      | 2355 |
| 7.154.2.5  | nppsSqrt_16u_ISfs . . . . .      | 2355 |
| 7.154.2.6  | nppsSqrt_16u_Sfs . . . . .       | 2356 |
| 7.154.2.7  | nppsSqrt_32f . . . . .           | 2356 |
| 7.154.2.8  | nppsSqrt_32f_I . . . . .         | 2356 |
| 7.154.2.9  | nppsSqrt_32fc . . . . .          | 2356 |
| 7.154.2.10 | nppsSqrt_32fc_I . . . . .        | 2357 |
| 7.154.2.11 | nppsSqrt_32s16s_Sfs . . . . .    | 2357 |
| 7.154.2.12 | nppsSqrt_64f . . . . .           | 2357 |
| 7.154.2.13 | nppsSqrt_64f_I . . . . .         | 2358 |
| 7.154.2.14 | nppsSqrt_64fc . . . . .          | 2358 |
| 7.154.2.15 | nppsSqrt_64fc_I . . . . .        | 2358 |
| 7.154.2.16 | nppsSqrt_64s16s_Sfs . . . . .    | 2358 |
| 7.154.2.17 | nppsSqrt_64s_ISfs . . . . .      | 2359 |
| 7.154.2.18 | nppsSqrt_64s_Sfs . . . . .       | 2359 |
| 7.154.2.19 | nppsSqrt_8u_ISfs . . . . .       | 2359 |
| 7.154.2.20 | nppsSqrt_8u_Sfs . . . . .        | 2359 |
| 7.155      | Cubrt . . . . .                  | 2361 |
| 7.155.1    | Detailed Description . . . . .   | 2361 |

|  |      |
|--|------|
| 7.155.2 Function Documentation . . . . . | 2361 |
| 7.155.2.1 nppsCubrt_32f . . . . .        | 2361 |
| 7.155.2.2 nppsCubrt_32s16s_Sfs . . . . . | 2361 |
| 7.156Exp . . . . .                       | 2362 |
| 7.156.1 Detailed Description . . . . .   | 2362 |
| 7.156.2 Function Documentation . . . . . | 2362 |
| 7.156.2.1 nppsExp_16s_ISfs . . . . .     | 2362 |
| 7.156.2.2 nppsExp_16s_Sfs . . . . .      | 2363 |
| 7.156.2.3 nppsExp_32f . . . . .          | 2363 |
| 7.156.2.4 nppsExp_32f64f . . . . .       | 2363 |
| 7.156.2.5 nppsExp_32f_I . . . . .        | 2364 |
| 7.156.2.6 nppsExp_32s_ISfs . . . . .     | 2364 |
| 7.156.2.7 nppsExp_32s_Sfs . . . . .      | 2364 |
| 7.156.2.8 nppsExp_64f . . . . .          | 2364 |
| 7.156.2.9 nppsExp_64f_I . . . . .        | 2365 |
| 7.156.2.10 nppsExp_64s_ISfs . . . . .    | 2365 |
| 7.156.2.11 nppsExp_64s_Sfs . . . . .     | 2365 |
| 7.157Ln . . . . .                        | 2366 |
| 7.157.1 Detailed Description . . . . .   | 2366 |
| 7.157.2 Function Documentation . . . . . | 2366 |
| 7.157.2.1 nppsLn_16s_ISfs . . . . .      | 2366 |
| 7.157.2.2 nppsLn_16s_Sfs . . . . .       | 2367 |
| 7.157.2.3 nppsLn_32f . . . . .           | 2367 |
| 7.157.2.4 nppsLn_32f_I . . . . .         | 2367 |
| 7.157.2.5 nppsLn_32s16s_Sfs . . . . .    | 2368 |
| 7.157.2.6 nppsLn_32s_ISfs . . . . .      | 2368 |
| 7.157.2.7 nppsLn_32s_Sfs . . . . .       | 2368 |
| 7.157.2.8 nppsLn_64f . . . . .           | 2369 |
| 7.157.2.9 nppsLn_64f32f . . . . .        | 2369 |
| 7.157.2.10 nppsLn_64f_I . . . . .        | 2369 |
| 7.15810Log10 . . . . .                   | 2370 |
| 7.158.1 Detailed Description . . . . .   | 2370 |
| 7.158.2 Function Documentation . . . . . | 2370 |
| 7.158.2.1 npps10Log10_32s_ISfs . . . . . | 2370 |
| 7.158.2.2 npps10Log10_32s_Sfs . . . . .  | 2370 |
| 7.159SumLn . . . . .                     | 2371 |

|   |      |
|---|------|
| 7.159.1 Detailed Description . . . . .            | 2371 |
| 7.159.2 Function Documentation . . . . .          | 2371 |
| 7.159.2.1 nppsSumLn_16s32f . . . . .              | 2371 |
| 7.159.2.2 nppsSumLn_32f . . . . .                 | 2372 |
| 7.159.2.3 nppsSumLn_32f64f . . . . .              | 2372 |
| 7.159.2.4 nppsSumLn_64f . . . . .                 | 2372 |
| 7.159.2.5 nppsSumLnGetBufferSize_16s32f . . . . . | 2373 |
| 7.159.2.6 nppsSumLnGetBufferSize_32f . . . . .    | 2373 |
| 7.159.2.7 nppsSumLnGetBufferSize_32f64f . . . . . | 2373 |
| 7.159.2.8 nppsSumLnGetBufferSize_64f . . . . .    | 2374 |
| 7.160Arctan . . . . .                             | 2375 |
| 7.160.1 Detailed Description . . . . .            | 2375 |
| 7.160.2 Function Documentation . . . . .          | 2375 |
| 7.160.2.1 nppsArctan_32f . . . . .                | 2375 |
| 7.160.2.2 nppsArctan_32f_I . . . . .              | 2375 |
| 7.160.2.3 nppsArctan_64f . . . . .                | 2376 |
| 7.160.2.4 nppsArctan_64f_I . . . . .              | 2376 |
| 7.161Normalize . . . . .                          | 2377 |
| 7.161.1 Detailed Description . . . . .            | 2377 |
| 7.161.2 Function Documentation . . . . .          | 2377 |
| 7.161.2.1 nppsNormalize_16s_Sfs . . . . .         | 2377 |
| 7.161.2.2 nppsNormalize_16sc_Sfs . . . . .        | 2378 |
| 7.161.2.3 nppsNormalize_32f . . . . .             | 2378 |
| 7.161.2.4 nppsNormalize_32fc . . . . .            | 2378 |
| 7.161.2.5 nppsNormalize_64f . . . . .             | 2379 |
| 7.161.2.6 nppsNormalize_64fc . . . . .            | 2379 |
| 7.162Cauchy, CauchyD, and CauchyDD2 . . . . .     | 2380 |
| 7.162.1 Detailed Description . . . . .            | 2380 |
| 7.162.2 Function Documentation . . . . .          | 2380 |
| 7.162.2.1 nppsCauchy_32f_I . . . . .              | 2380 |
| 7.162.2.2 nppsCauchyD_32f_I . . . . .             | 2380 |
| 7.162.2.3 nppsCauchyDD2_32f_I . . . . .           | 2381 |
| 7.163Logical And Shift Operations . . . . .       | 2382 |
| 7.164AndC . . . . .                               | 2383 |
| 7.164.1 Detailed Description . . . . .            | 2383 |
| 7.164.2 Function Documentation . . . . .          | 2383 |

|  |      |
|--|------|
| 7.164.2.1 nppsAndC_16u . . . . .         | 2383 |
| 7.164.2.2 nppsAndC_16u_I . . . . .       | 2384 |
| 7.164.2.3 nppsAndC_32u . . . . .         | 2384 |
| 7.164.2.4 nppsAndC_32u_I . . . . .       | 2384 |
| 7.164.2.5 nppsAndC_8u . . . . .          | 2384 |
| 7.164.2.6 nppsAndC_8u_I . . . . .        | 2385 |
| 7.165And . . . . .                       | 2386 |
| 7.165.1 Detailed Description . . . . .   | 2386 |
| 7.165.2 Function Documentation . . . . . | 2386 |
| 7.165.2.1 nppsAnd_16u . . . . .          | 2386 |
| 7.165.2.2 nppsAnd_16u_I . . . . .        | 2387 |
| 7.165.2.3 nppsAnd_32u . . . . .          | 2387 |
| 7.165.2.4 nppsAnd_32u_I . . . . .        | 2387 |
| 7.165.2.5 nppsAnd_8u . . . . .           | 2387 |
| 7.165.2.6 nppsAnd_8u_I . . . . .         | 2388 |
| 7.166OrC . . . . .                       | 2389 |
| 7.166.1 Detailed Description . . . . .   | 2389 |
| 7.166.2 Function Documentation . . . . . | 2389 |
| 7.166.2.1 nppsOrC_16u . . . . .          | 2389 |
| 7.166.2.2 nppsOrC_16u_I . . . . .        | 2390 |
| 7.166.2.3 nppsOrC_32u . . . . .          | 2390 |
| 7.166.2.4 nppsOrC_32u_I . . . . .        | 2390 |
| 7.166.2.5 nppsOrC_8u . . . . .           | 2390 |
| 7.166.2.6 nppsOrC_8u_I . . . . .         | 2391 |
| 7.167Or . . . . .                        | 2392 |
| 7.167.1 Detailed Description . . . . .   | 2392 |
| 7.167.2 Function Documentation . . . . . | 2392 |
| 7.167.2.1 nppsOr_16u . . . . .           | 2392 |
| 7.167.2.2 nppsOr_16u_I . . . . .         | 2393 |
| 7.167.2.3 nppsOr_32u . . . . .           | 2393 |
| 7.167.2.4 nppsOr_32u_I . . . . .         | 2393 |
| 7.167.2.5 nppsOr_8u . . . . .            | 2393 |
| 7.167.2.6 nppsOr_8u_I . . . . .          | 2394 |
| 7.168XorC . . . . .                      | 2395 |
| 7.168.1 Detailed Description . . . . .   | 2395 |
| 7.168.2 Function Documentation . . . . . | 2395 |



|  |      |
|--|------|
| 7.168.2.1 nppsXorC_16u . . . . .         | 2395 |
| 7.168.2.2 nppsXorC_16u_I . . . . .       | 2396 |
| 7.168.2.3 nppsXorC_32u . . . . .         | 2396 |
| 7.168.2.4 nppsXorC_32u_I . . . . .       | 2396 |
| 7.168.2.5 nppsXorC_8u . . . . .          | 2396 |
| 7.168.2.6 nppsXorC_8u_I . . . . .        | 2397 |
| 7.169Xor . . . . .                       | 2398 |
| 7.169.1 Detailed Description . . . . .   | 2398 |
| 7.169.2 Function Documentation . . . . . | 2398 |
| 7.169.2.1 nppsXor_16u . . . . .          | 2398 |
| 7.169.2.2 nppsXor_16u_I . . . . .        | 2399 |
| 7.169.2.3 nppsXor_32u . . . . .          | 2399 |
| 7.169.2.4 nppsXor_32u_I . . . . .        | 2399 |
| 7.169.2.5 nppsXor_8u . . . . .           | 2399 |
| 7.169.2.6 nppsXor_8u_I . . . . .         | 2400 |
| 7.170Not . . . . .                       | 2401 |
| 7.170.1 Detailed Description . . . . .   | 2401 |
| 7.170.2 Function Documentation . . . . . | 2401 |
| 7.170.2.1 nppsNot_16u . . . . .          | 2401 |
| 7.170.2.2 nppsNot_16u_I . . . . .        | 2402 |
| 7.170.2.3 nppsNot_32u . . . . .          | 2402 |
| 7.170.2.4 nppsNot_32u_I . . . . .        | 2402 |
| 7.170.2.5 nppsNot_8u . . . . .           | 2402 |
| 7.170.2.6 nppsNot_8u_I . . . . .         | 2403 |
| 7.171LShiftC . . . . .                   | 2404 |
| 7.171.1 Detailed Description . . . . .   | 2404 |
| 7.171.2 Function Documentation . . . . . | 2404 |
| 7.171.2.1 nppsLShiftC_16s . . . . .      | 2404 |
| 7.171.2.2 nppsLShiftC_16s_I . . . . .    | 2405 |
| 7.171.2.3 nppsLShiftC_16u . . . . .      | 2405 |
| 7.171.2.4 nppsLShiftC_16u_I . . . . .    | 2405 |
| 7.171.2.5 nppsLShiftC_32s . . . . .      | 2406 |
| 7.171.2.6 nppsLShiftC_32s_I . . . . .    | 2406 |
| 7.171.2.7 nppsLShiftC_32u . . . . .      | 2406 |
| 7.171.2.8 nppsLShiftC_32u_I . . . . .    | 2407 |
| 7.171.2.9 nppsLShiftC_8u . . . . .       | 2407 |

|            |                                  |      |
|------------|----------------------------------|------|
| 7.171.2.10 | nppsLShiftC_8u_I . . . . .       | 2407 |
| 7.172      | RShiftC . . . . .                | 2408 |
| 7.172.1    | Detailed Description . . . . .   | 2408 |
| 7.172.2    | Function Documentation . . . . . | 2408 |
| 7.172.2.1  | nppsRShiftC_16s . . . . .        | 2408 |
| 7.172.2.2  | nppsRShiftC_16s_I . . . . .      | 2409 |
| 7.172.2.3  | nppsRShiftC_16u . . . . .        | 2409 |
| 7.172.2.4  | nppsRShiftC_16u_I . . . . .      | 2409 |
| 7.172.2.5  | nppsRShiftC_32s . . . . .        | 2410 |
| 7.172.2.6  | nppsRShiftC_32s_I . . . . .      | 2410 |
| 7.172.2.7  | nppsRShiftC_32u . . . . .        | 2410 |
| 7.172.2.8  | nppsRShiftC_32u_I . . . . .      | 2411 |
| 7.172.2.9  | nppsRShiftC_8u . . . . .         | 2411 |
| 7.172.2.10 | nppsRShiftC_8u_I . . . . .       | 2411 |
| 7.173      | Conversion Functions . . . . .   | 2412 |
| 7.174      | Convert . . . . .                | 2413 |
| 7.174.1    | Function Documentation . . . . . | 2415 |
| 7.174.1.1  | nppsConvert_16s32f . . . . .     | 2415 |
| 7.174.1.2  | nppsConvert_16s32f_Sfs . . . . . | 2415 |
| 7.174.1.3  | nppsConvert_16s32s . . . . .     | 2415 |
| 7.174.1.4  | nppsConvert_16s64f_Sfs . . . . . | 2415 |
| 7.174.1.5  | nppsConvert_16s8s_Sfs . . . . .  | 2415 |
| 7.174.1.6  | nppsConvert_16u32f . . . . .     | 2415 |
| 7.174.1.7  | nppsConvert_32f16s_Sfs . . . . . | 2415 |
| 7.174.1.8  | nppsConvert_32f16u_Sfs . . . . . | 2415 |
| 7.174.1.9  | nppsConvert_32f32s_Sfs . . . . . | 2415 |
| 7.174.1.10 | nppsConvert_32f64f . . . . .     | 2415 |
| 7.174.1.11 | nppsConvert_32f8s_Sfs . . . . .  | 2415 |
| 7.174.1.12 | nppsConvert_32f8u_Sfs . . . . .  | 2415 |
| 7.174.1.13 | nppsConvert_32s16s . . . . .     | 2415 |
| 7.174.1.14 | nppsConvert_32s16s_Sfs . . . . . | 2415 |
| 7.174.1.15 | nppsConvert_32s32f . . . . .     | 2415 |
| 7.174.1.16 | nppsConvert_32s32f_Sfs . . . . . | 2415 |
| 7.174.1.17 | nppsConvert_32s64f . . . . .     | 2415 |
| 7.174.1.18 | nppsConvert_32s64f_Sfs . . . . . | 2415 |
| 7.174.1.19 | nppsConvert_64f16s_Sfs . . . . . | 2415 |

|            |                                     |      |
|------------|-------------------------------------|------|
| 7.174.1.20 | nppsConvert_64f32f . . . . .        | 2415 |
| 7.174.1.21 | nppsConvert_64f32s_Sfs . . . . .    | 2415 |
| 7.174.1.22 | nppsConvert_64f64s_Sfs . . . . .    | 2415 |
| 7.174.1.23 | nppsConvert_64s32s_Sfs . . . . .    | 2415 |
| 7.174.1.24 | nppsConvert_64s64f . . . . .        | 2415 |
| 7.174.1.25 | nppsConvert_8s16s . . . . .         | 2415 |
| 7.174.1.26 | nppsConvert_8s32f . . . . .         | 2415 |
| 7.174.1.27 | nppsConvert_8u32f . . . . .         | 2415 |
| 7.175      | Threshold . . . . .                 | 2416 |
| 7.175.1    | Function Documentation . . . . .    | 2420 |
| 7.175.1.1  | nppsThreshold_16s . . . . .         | 2420 |
| 7.175.1.2  | nppsThreshold_16s_I . . . . .       | 2421 |
| 7.175.1.3  | nppsThreshold_16sc . . . . .        | 2421 |
| 7.175.1.4  | nppsThreshold_16sc_I . . . . .      | 2421 |
| 7.175.1.5  | nppsThreshold_32f . . . . .         | 2422 |
| 7.175.1.6  | nppsThreshold_32f_I . . . . .       | 2422 |
| 7.175.1.7  | nppsThreshold_32fc . . . . .        | 2422 |
| 7.175.1.8  | nppsThreshold_32fc_I . . . . .      | 2423 |
| 7.175.1.9  | nppsThreshold_64f . . . . .         | 2423 |
| 7.175.1.10 | nppsThreshold_64f_I . . . . .       | 2424 |
| 7.175.1.11 | nppsThreshold_64fc . . . . .        | 2424 |
| 7.175.1.12 | nppsThreshold_64fc_I . . . . .      | 2424 |
| 7.175.1.13 | nppsThreshold_GT_16s . . . . .      | 2425 |
| 7.175.1.14 | nppsThreshold_GT_16s_I . . . . .    | 2425 |
| 7.175.1.15 | nppsThreshold_GT_16sc . . . . .     | 2425 |
| 7.175.1.16 | nppsThreshold_GT_16sc_I . . . . .   | 2426 |
| 7.175.1.17 | nppsThreshold_GT_32f . . . . .      | 2426 |
| 7.175.1.18 | nppsThreshold_GT_32f_I . . . . .    | 2426 |
| 7.175.1.19 | nppsThreshold_GT_32fc . . . . .     | 2427 |
| 7.175.1.20 | nppsThreshold_GT_32fc_I . . . . .   | 2427 |
| 7.175.1.21 | nppsThreshold_GT_64f . . . . .      | 2427 |
| 7.175.1.22 | nppsThreshold_GT_64f_I . . . . .    | 2428 |
| 7.175.1.23 | nppsThreshold_GT_64fc . . . . .     | 2428 |
| 7.175.1.24 | nppsThreshold_GT_64fc_I . . . . .   | 2428 |
| 7.175.1.25 | nppsThreshold_GTVa1_16s . . . . .   | 2429 |
| 7.175.1.26 | nppsThreshold_GTVa1_16s_I . . . . . | 2429 |

|            |                                      |      |
|------------|--------------------------------------|------|
| 7.175.1.27 | nppsThreshold_GTVal_16sc . . . . .   | 2429 |
| 7.175.1.28 | nppsThreshold_GTVal_16sc_I . . . . . | 2430 |
| 7.175.1.29 | nppsThreshold_GTVal_32f . . . . .    | 2430 |
| 7.175.1.30 | nppsThreshold_GTVal_32f_I . . . . .  | 2430 |
| 7.175.1.31 | nppsThreshold_GTVal_32fc . . . . .   | 2431 |
| 7.175.1.32 | nppsThreshold_GTVal_32fc_I . . . . . | 2431 |
| 7.175.1.33 | nppsThreshold_GTVal_64f . . . . .    | 2431 |
| 7.175.1.34 | nppsThreshold_GTVal_64f_I . . . . .  | 2432 |
| 7.175.1.35 | nppsThreshold_GTVal_64fc . . . . .   | 2432 |
| 7.175.1.36 | nppsThreshold_GTVal_64fc_I . . . . . | 2432 |
| 7.175.1.37 | nppsThreshold_LT_16s . . . . .       | 2433 |
| 7.175.1.38 | nppsThreshold_LT_16s_I . . . . .     | 2433 |
| 7.175.1.39 | nppsThreshold_LT_16sc . . . . .      | 2433 |
| 7.175.1.40 | nppsThreshold_LT_16sc_I . . . . .    | 2434 |
| 7.175.1.41 | nppsThreshold_LT_32f . . . . .       | 2434 |
| 7.175.1.42 | nppsThreshold_LT_32f_I . . . . .     | 2434 |
| 7.175.1.43 | nppsThreshold_LT_32fc . . . . .      | 2435 |
| 7.175.1.44 | nppsThreshold_LT_32fc_I . . . . .    | 2435 |
| 7.175.1.45 | nppsThreshold_LT_64f . . . . .       | 2435 |
| 7.175.1.46 | nppsThreshold_LT_64f_I . . . . .     | 2436 |
| 7.175.1.47 | nppsThreshold_LT_64fc . . . . .      | 2436 |
| 7.175.1.48 | nppsThreshold_LT_64fc_I . . . . .    | 2436 |
| 7.175.1.49 | nppsThreshold_LTVal_16s . . . . .    | 2437 |
| 7.175.1.50 | nppsThreshold_LTVal_16s_I . . . . .  | 2437 |
| 7.175.1.51 | nppsThreshold_LTVal_16sc . . . . .   | 2437 |
| 7.175.1.52 | nppsThreshold_LTVal_16sc_I . . . . . | 2438 |
| 7.175.1.53 | nppsThreshold_LTVal_32f . . . . .    | 2438 |
| 7.175.1.54 | nppsThreshold_LTVal_32f_I . . . . .  | 2438 |
| 7.175.1.55 | nppsThreshold_LTVal_32fc . . . . .   | 2439 |
| 7.175.1.56 | nppsThreshold_LTVal_32fc_I . . . . . | 2439 |
| 7.175.1.57 | nppsThreshold_LTVal_64f . . . . .    | 2439 |
| 7.175.1.58 | nppsThreshold_LTVal_64f_I . . . . .  | 2440 |
| 7.175.1.59 | nppsThreshold_LTVal_64fc . . . . .   | 2440 |
| 7.175.1.60 | nppsThreshold_LTVal_64fc_I . . . . . | 2440 |
| 7.176      | Filtering Functions . . . . .        | 2441 |
| 7.176.1    | Detailed Description . . . . .       | 2441 |

|   |      |
|---|------|
| 7.177Integral                           | 2442 |
| 7.177.1 Detailed Description            | 2442 |
| 7.177.2 Function Documentation          | 2442 |
| 7.177.2.1 nppsIntegral_32s              | 2442 |
| 7.177.2.2 nppsIntegralGetBufferSize_32s | 2442 |
| 7.178Initialization                     | 2443 |
| 7.179Set                                | 2444 |
| 7.179.1 Function Documentation          | 2445 |
| 7.179.1.1 nppsSet_16s                   | 2445 |
| 7.179.1.2 nppsSet_16sc                  | 2445 |
| 7.179.1.3 nppsSet_16u                   | 2445 |
| 7.179.1.4 nppsSet_32f                   | 2445 |
| 7.179.1.5 nppsSet_32fc                  | 2446 |
| 7.179.1.6 nppsSet_32s                   | 2446 |
| 7.179.1.7 nppsSet_32sc                  | 2446 |
| 7.179.1.8 nppsSet_32u                   | 2447 |
| 7.179.1.9 nppsSet_64f                   | 2447 |
| 7.179.1.10nppsSet_64fc                  | 2447 |
| 7.179.1.11nppsSet_64s                   | 2447 |
| 7.179.1.12nppsSet_64sc                  | 2448 |
| 7.179.1.13nppsSet_8s                    | 2448 |
| 7.179.1.14nppsSet_8u                    | 2448 |
| 7.180Zero                               | 2449 |
| 7.180.1 Function Documentation          | 2449 |
| 7.180.1.1 nppsZero_16s                  | 2449 |
| 7.180.1.2 nppsZero_16sc                 | 2450 |
| 7.180.1.3 nppsZero_32f                  | 2450 |
| 7.180.1.4 nppsZero_32fc                 | 2450 |
| 7.180.1.5 nppsZero_32s                  | 2450 |
| 7.180.1.6 nppsZero_32sc                 | 2451 |
| 7.180.1.7 nppsZero_64f                  | 2451 |
| 7.180.1.8 nppsZero_64fc                 | 2451 |
| 7.180.1.9 nppsZero_64s                  | 2451 |
| 7.180.1.10nppsZero_64sc                 | 2452 |
| 7.180.1.11nppsZero_8u                   | 2452 |
| 7.181Copy                               | 2453 |

|   |      |
|---|------|
| 7.181.1 Function Documentation . . . . .        | 2453 |
| 7.181.1.1 nppsCopy_16s . . . . .                | 2453 |
| 7.181.1.2 nppsCopy_16sc . . . . .               | 2454 |
| 7.181.1.3 nppsCopy_32f . . . . .                | 2454 |
| 7.181.1.4 nppsCopy_32fc . . . . .               | 2454 |
| 7.181.1.5 nppsCopy_32s . . . . .                | 2455 |
| 7.181.1.6 nppsCopy_32sc . . . . .               | 2455 |
| 7.181.1.7 nppsCopy_64fc . . . . .               | 2455 |
| 7.181.1.8 nppsCopy_64s . . . . .                | 2455 |
| 7.181.1.9 nppsCopy_64sc . . . . .               | 2456 |
| 7.181.1.10 nppsCopy_8u . . . . .                | 2456 |
| 7.182 Statistical Functions . . . . .           | 2457 |
| 7.182.1 Detailed Description . . . . .          | 2457 |
| 7.183 MinEvery And MaxEvery Functions . . . . . | 2458 |
| 7.183.1 Detailed Description . . . . .          | 2458 |
| 7.183.2 Function Documentation . . . . .        | 2458 |
| 7.183.2.1 nppsMaxEvery_16s_I . . . . .          | 2458 |
| 7.183.2.2 nppsMaxEvery_16u_I . . . . .          | 2459 |
| 7.183.2.3 nppsMaxEvery_32f_I . . . . .          | 2459 |
| 7.183.2.4 nppsMaxEvery_32s_I . . . . .          | 2459 |
| 7.183.2.5 nppsMaxEvery_8u_I . . . . .           | 2460 |
| 7.183.2.6 nppsMinEvery_16s_I . . . . .          | 2460 |
| 7.183.2.7 nppsMinEvery_16u_I . . . . .          | 2460 |
| 7.183.2.8 nppsMinEvery_32f_I . . . . .          | 2460 |
| 7.183.2.9 nppsMinEvery_32s_I . . . . .          | 2461 |
| 7.183.2.10 nppsMinEvery_64f_I . . . . .         | 2461 |
| 7.183.2.11 nppsMinEvery_8u_I . . . . .          | 2461 |
| 7.184 Sum . . . . .                             | 2462 |
| 7.184.1 Detailed Description . . . . .          | 2463 |
| 7.184.2 Function Documentation . . . . .        | 2463 |
| 7.184.2.1 nppsSum_16s32s_Sfs . . . . .          | 2463 |
| 7.184.2.2 nppsSum_16s_Sfs . . . . .             | 2463 |
| 7.184.2.3 nppsSum_16sc32sc_Sfs . . . . .        | 2464 |
| 7.184.2.4 nppsSum_16sc_Sfs . . . . .            | 2464 |
| 7.184.2.5 nppsSum_32f . . . . .                 | 2465 |
| 7.184.2.6 nppsSum_32fc . . . . .                | 2465 |

|            |                                   |      |
|------------|-----------------------------------|------|
| 7.184.2.7  | nppsSum_32s_Sfs                   | 2465 |
| 7.184.2.8  | nppsSum_64f                       | 2466 |
| 7.184.2.9  | nppsSum_64fc                      | 2466 |
| 7.184.2.10 | nppsSumGetBufferSize_16s32s_Sfs   | 2466 |
| 7.184.2.11 | nppsSumGetBufferSize_16s_Sfs      | 2467 |
| 7.184.2.12 | nppsSumGetBufferSize_16sc32sc_Sfs | 2467 |
| 7.184.2.13 | nppsSumGetBufferSize_16sc_Sfs     | 2467 |
| 7.184.2.14 | nppsSumGetBufferSize_32f          | 2467 |
| 7.184.2.15 | nppsSumGetBufferSize_32fc         | 2468 |
| 7.184.2.16 | nppsSumGetBufferSize_32s_Sfs      | 2468 |
| 7.184.2.17 | nppsSumGetBufferSize_64f          | 2468 |
| 7.184.2.18 | nppsSumGetBufferSize_64fc         | 2468 |
| 7.185      | Maximum                           | 2469 |
| 7.185.1    | Function Documentation            | 2470 |
| 7.185.1.1  | nppsMax_16s                       | 2470 |
| 7.185.1.2  | nppsMax_32f                       | 2471 |
| 7.185.1.3  | nppsMax_32s                       | 2471 |
| 7.185.1.4  | nppsMax_64f                       | 2471 |
| 7.185.1.5  | nppsMaxAbs_16s                    | 2472 |
| 7.185.1.6  | nppsMaxAbs_32s                    | 2472 |
| 7.185.1.7  | nppsMaxAbsGetBufferSize_16s       | 2472 |
| 7.185.1.8  | nppsMaxAbsGetBufferSize_32s       | 2473 |
| 7.185.1.9  | nppsMaxAbsIndx_16s                | 2473 |
| 7.185.1.10 | nppsMaxAbsIndx_32s                | 2473 |
| 7.185.1.11 | nppsMaxAbsIndxGetBufferSize_16s   | 2474 |
| 7.185.1.12 | nppsMaxAbsIndxGetBufferSize_32s   | 2474 |
| 7.185.1.13 | nppsMaxGetBufferSize_16s          | 2474 |
| 7.185.1.14 | nppsMaxGetBufferSize_32f          | 2475 |
| 7.185.1.15 | nppsMaxGetBufferSize_32s          | 2475 |
| 7.185.1.16 | nppsMaxGetBufferSize_64f          | 2475 |
| 7.185.1.17 | nppsMaxIndx_16s                   | 2475 |
| 7.185.1.18 | nppsMaxIndx_32f                   | 2476 |
| 7.185.1.19 | nppsMaxIndx_32s                   | 2476 |
| 7.185.1.20 | nppsMaxIndx_64f                   | 2477 |
| 7.185.1.21 | nppsMaxIndxGetBufferSize_16s      | 2477 |
| 7.185.1.22 | nppsMaxIndxGetBufferSize_32f      | 2477 |

|            |                                 |      |
|------------|---------------------------------|------|
| 7.185.1.23 | nppsMaxIndxGetBufferSize_32s    | 2478 |
| 7.185.1.24 | nppsMaxIndxGetBufferSize_64f    | 2478 |
| 7.186      | Minimum                         | 2479 |
| 7.186.1    | Function Documentation          | 2480 |
| 7.186.1.1  | nppsMin_16s                     | 2480 |
| 7.186.1.2  | nppsMin_32f                     | 2481 |
| 7.186.1.3  | nppsMin_32s                     | 2481 |
| 7.186.1.4  | nppsMin_64f                     | 2481 |
| 7.186.1.5  | nppsMinAbs_16s                  | 2482 |
| 7.186.1.6  | nppsMinAbs_32s                  | 2482 |
| 7.186.1.7  | nppsMinAbsGetBufferSize_16s     | 2482 |
| 7.186.1.8  | nppsMinAbsGetBufferSize_32s     | 2483 |
| 7.186.1.9  | nppsMinAbsIndx_16s              | 2483 |
| 7.186.1.10 | nppsMinAbsIndx_32s              | 2483 |
| 7.186.1.11 | nppsMinAbsIndxGetBufferSize_16s | 2484 |
| 7.186.1.12 | nppsMinAbsIndxGetBufferSize_32s | 2484 |
| 7.186.1.13 | nppsMinGetBufferSize_16s        | 2484 |
| 7.186.1.14 | nppsMinGetBufferSize_32f        | 2485 |
| 7.186.1.15 | nppsMinGetBufferSize_32s        | 2485 |
| 7.186.1.16 | nppsMinGetBufferSize_64f        | 2485 |
| 7.186.1.17 | nppsMinIndx_16s                 | 2485 |
| 7.186.1.18 | nppsMinIndx_32f                 | 2486 |
| 7.186.1.19 | nppsMinIndx_32s                 | 2486 |
| 7.186.1.20 | nppsMinIndx_64f                 | 2487 |
| 7.186.1.21 | nppsMinIndxGetBufferSize_16s    | 2487 |
| 7.186.1.22 | nppsMinIndxGetBufferSize_32f    | 2487 |
| 7.186.1.23 | nppsMinIndxGetBufferSize_32s    | 2488 |
| 7.186.1.24 | nppsMinIndxGetBufferSize_64f    | 2488 |
| 7.187      | Mean                            | 2489 |
| 7.187.1    | Function Documentation          | 2490 |
| 7.187.1.1  | nppsMean_16s_Sfs                | 2490 |
| 7.187.1.2  | nppsMean_16sc_Sfs               | 2490 |
| 7.187.1.3  | nppsMean_32f                    | 2490 |
| 7.187.1.4  | nppsMean_32fc                   | 2491 |
| 7.187.1.5  | nppsMean_32s_Sfs                | 2491 |
| 7.187.1.6  | nppsMean_64f                    | 2492 |



|  |      |
|--|------|
| 7.187.1.7 nppsMean_64fc . . . . .                          | 2492 |
| 7.187.1.8 nppsMeanGetBufferSize_16s_Sfs . . . . .          | 2492 |
| 7.187.1.9 nppsMeanGetBufferSize_16sc_Sfs . . . . .         | 2493 |
| 7.187.1.10 nppsMeanGetBufferSize_32f . . . . .             | 2493 |
| 7.187.1.11 nppsMeanGetBufferSize_32fc . . . . .            | 2493 |
| 7.187.1.12 nppsMeanGetBufferSize_32s_Sfs . . . . .         | 2493 |
| 7.187.1.13 nppsMeanGetBufferSize_64f . . . . .             | 2494 |
| 7.187.1.14 nppsMeanGetBufferSize_64fc . . . . .            | 2494 |
| 7.188 Standard Deviation . . . . .                         | 2495 |
| 7.188.1 Function Documentation . . . . .                   | 2495 |
| 7.188.1.1 nppsStdDev_16s32s_Sfs . . . . .                  | 2495 |
| 7.188.1.2 nppsStdDev_16s_Sfs . . . . .                     | 2496 |
| 7.188.1.3 nppsStdDev_32f . . . . .                         | 2496 |
| 7.188.1.4 nppsStdDev_64f . . . . .                         | 2496 |
| 7.188.1.5 nppsStdDevGetBufferSize_16s32s_Sfs . . . . .     | 2497 |
| 7.188.1.6 nppsStdDevGetBufferSize_16s_Sfs . . . . .        | 2497 |
| 7.188.1.7 nppsStdDevGetBufferSize_32f . . . . .            | 2497 |
| 7.188.1.8 nppsStdDevGetBufferSize_64f . . . . .            | 2497 |
| 7.189 Mean And Standard Deviation . . . . .                | 2498 |
| 7.189.1 Function Documentation . . . . .                   | 2498 |
| 7.189.1.1 nppsMeanStdDev_16s32s_Sfs . . . . .              | 2498 |
| 7.189.1.2 nppsMeanStdDev_16s_Sfs . . . . .                 | 2499 |
| 7.189.1.3 nppsMeanStdDev_32f . . . . .                     | 2499 |
| 7.189.1.4 nppsMeanStdDev_64f . . . . .                     | 2499 |
| 7.189.1.5 nppsMeanStdDevGetBufferSize_16s32s_Sfs . . . . . | 2500 |
| 7.189.1.6 nppsMeanStdDevGetBufferSize_16s_Sfs . . . . .    | 2500 |
| 7.189.1.7 nppsMeanStdDevGetBufferSize_32f . . . . .        | 2500 |
| 7.189.1.8 nppsMeanStdDevGetBufferSize_64f . . . . .        | 2501 |
| 7.190 Minimum_Maximum . . . . .                            | 2502 |
| 7.190.1 Function Documentation . . . . .                   | 2504 |
| 7.190.1.1 nppsMinMax_16s . . . . .                         | 2504 |
| 7.190.1.2 nppsMinMax_16u . . . . .                         | 2504 |
| 7.190.1.3 nppsMinMax_32f . . . . .                         | 2504 |
| 7.190.1.4 nppsMinMax_32s . . . . .                         | 2505 |
| 7.190.1.5 nppsMinMax_32u . . . . .                         | 2505 |
| 7.190.1.6 nppsMinMax_64f . . . . .                         | 2505 |

|   |      |
|---|------|
| 7.190.1.7 nppsMinMax_8u . . . . .                       | 2506 |
| 7.190.1.8 nppsMinMaxGetBufferSize_16s . . . . .         | 2506 |
| 7.190.1.9 nppsMinMaxGetBufferSize_16u . . . . .         | 2506 |
| 7.190.1.10 nppsMinMaxGetBufferSize_32f . . . . .        | 2507 |
| 7.190.1.11 nppsMinMaxGetBufferSize_32s . . . . .        | 2507 |
| 7.190.1.12 nppsMinMaxGetBufferSize_32u . . . . .        | 2507 |
| 7.190.1.13 nppsMinMaxGetBufferSize_64f . . . . .        | 2508 |
| 7.190.1.14 nppsMinMaxGetBufferSize_8u . . . . .         | 2508 |
| 7.190.1.15 nppsMinMaxIndx_16s . . . . .                 | 2508 |
| 7.190.1.16 nppsMinMaxIndx_16u . . . . .                 | 2509 |
| 7.190.1.17 nppsMinMaxIndx_32f . . . . .                 | 2509 |
| 7.190.1.18 nppsMinMaxIndx_32s . . . . .                 | 2509 |
| 7.190.1.19 nppsMinMaxIndx_32u . . . . .                 | 2510 |
| 7.190.1.20 nppsMinMaxIndx_64f . . . . .                 | 2510 |
| 7.190.1.21 nppsMinMaxIndx_8u . . . . .                  | 2511 |
| 7.190.1.22 nppsMinMaxIndxGetBufferSize_16s . . . . .    | 2511 |
| 7.190.1.23 nppsMinMaxIndxGetBufferSize_16u . . . . .    | 2511 |
| 7.190.1.24 nppsMinMaxIndxGetBufferSize_32f . . . . .    | 2512 |
| 7.190.1.25 nppsMinMaxIndxGetBufferSize_32s . . . . .    | 2512 |
| 7.190.1.26 nppsMinMaxIndxGetBufferSize_32u . . . . .    | 2512 |
| 7.190.1.27 nppsMinMaxIndxGetBufferSize_64f . . . . .    | 2512 |
| 7.190.1.28 nppsMinMaxIndxGetBufferSize_8u . . . . .     | 2513 |
| 7.191 Infinity Norm . . . . .                           | 2514 |
| 7.191.1 Function Documentation . . . . .                | 2515 |
| 7.191.1.1 nppsNorm_Inf_16s32f . . . . .                 | 2515 |
| 7.191.1.2 nppsNorm_Inf_16s32s_Sfs . . . . .             | 2515 |
| 7.191.1.3 nppsNorm_Inf_32f . . . . .                    | 2515 |
| 7.191.1.4 nppsNorm_Inf_32fc32f . . . . .                | 2516 |
| 7.191.1.5 nppsNorm_Inf_64f . . . . .                    | 2516 |
| 7.191.1.6 nppsNorm_Inf_64fc64f . . . . .                | 2516 |
| 7.191.1.7 nppsNormInfGetBufferSize_16s32f . . . . .     | 2517 |
| 7.191.1.8 nppsNormInfGetBufferSize_16s32s_Sfs . . . . . | 2517 |
| 7.191.1.9 nppsNormInfGetBufferSize_32f . . . . .        | 2517 |
| 7.191.1.10 nppsNormInfGetBufferSize_32fc32f . . . . .   | 2517 |
| 7.191.1.11 nppsNormInfGetBufferSize_64f . . . . .       | 2518 |
| 7.191.1.12 nppsNormInfGetBufferSize_64fc64f . . . . .   | 2518 |

|   |      |
|---|------|
| 7.192L1 Norm . . . . .                                    | 2519 |
| 7.192.1 Function Documentation . . . . .                  | 2520 |
| 7.192.1.1 nppsNorm_L1_16s32f . . . . .                    | 2520 |
| 7.192.1.2 nppsNorm_L1_16s32s_Sfs . . . . .                | 2520 |
| 7.192.1.3 nppsNorm_L1_16s64s_Sfs . . . . .                | 2520 |
| 7.192.1.4 nppsNorm_L1_32f . . . . .                       | 2521 |
| 7.192.1.5 nppsNorm_L1_32fc64f . . . . .                   | 2521 |
| 7.192.1.6 nppsNorm_L1_64f . . . . .                       | 2521 |
| 7.192.1.7 nppsNorm_L1_64fc64f . . . . .                   | 2522 |
| 7.192.1.8 nppsNormL1GetBufferSize_16s32f . . . . .        | 2522 |
| 7.192.1.9 nppsNormL1GetBufferSize_16s32s_Sfs . . . . .    | 2522 |
| 7.192.1.10nppsNormL1GetBufferSize_16s64s_Sfs . . . . .    | 2523 |
| 7.192.1.11nppsNormL1GetBufferSize_32f . . . . .           | 2523 |
| 7.192.1.12nppsNormL1GetBufferSize_32fc64f . . . . .       | 2523 |
| 7.192.1.13nppsNormL1GetBufferSize_64f . . . . .           | 2523 |
| 7.192.1.14nppsNormL1GetBufferSize_64fc64f . . . . .       | 2524 |
| 7.193L2 Norm . . . . .                                    | 2525 |
| 7.193.1 Function Documentation . . . . .                  | 2526 |
| 7.193.1.1 nppsNorm_L2_16s32f . . . . .                    | 2526 |
| 7.193.1.2 nppsNorm_L2_16s32s_Sfs . . . . .                | 2526 |
| 7.193.1.3 nppsNorm_L2_32f . . . . .                       | 2526 |
| 7.193.1.4 nppsNorm_L2_32fc64f . . . . .                   | 2527 |
| 7.193.1.5 nppsNorm_L2_64f . . . . .                       | 2527 |
| 7.193.1.6 nppsNorm_L2_64fc64f . . . . .                   | 2527 |
| 7.193.1.7 nppsNorm_L2Sqr_16s64s_Sfs . . . . .             | 2528 |
| 7.193.1.8 nppsNormL2GetBufferSize_16s32f . . . . .        | 2528 |
| 7.193.1.9 nppsNormL2GetBufferSize_16s32s_Sfs . . . . .    | 2528 |
| 7.193.1.10nppsNormL2GetBufferSize_32f . . . . .           | 2529 |
| 7.193.1.11nppsNormL2GetBufferSize_32fc64f . . . . .       | 2529 |
| 7.193.1.12nppsNormL2GetBufferSize_64f . . . . .           | 2529 |
| 7.193.1.13nppsNormL2GetBufferSize_64fc64f . . . . .       | 2529 |
| 7.193.1.14nppsNormL2SqrGetBufferSize_16s64s_Sfs . . . . . | 2530 |
| 7.194Infinity Norm Diff . . . . .                         | 2531 |
| 7.194.1 Function Documentation . . . . .                  | 2532 |
| 7.194.1.1 nppsNormDiff_Inf_16s32f . . . . .               | 2532 |
| 7.194.1.2 nppsNormDiff_Inf_16s32s_Sfs . . . . .           | 2532 |

|   |      |
|---|------|
| 7.194.1.3 nppsNormDiff_Inf_32f . . . . .                    | 2532 |
| 7.194.1.4 nppsNormDiff_Inf_32fc32f . . . . .                | 2533 |
| 7.194.1.5 nppsNormDiff_Inf_64f . . . . .                    | 2533 |
| 7.194.1.6 nppsNormDiff_Inf_64fc64f . . . . .                | 2534 |
| 7.194.1.7 nppsNormDiffInfGetBufferSize_16s32f . . . . .     | 2534 |
| 7.194.1.8 nppsNormDiffInfGetBufferSize_16s32s_Sfs . . . . . | 2534 |
| 7.194.1.9 nppsNormDiffInfGetBufferSize_32f . . . . .        | 2534 |
| 7.194.1.10 nppsNormDiffInfGetBufferSize_32fc32f . . . . .   | 2535 |
| 7.194.1.11 nppsNormDiffInfGetBufferSize_64f . . . . .       | 2535 |
| 7.194.1.12 nppsNormDiffInfGetBufferSize_64fc64f . . . . .   | 2535 |
| 7.195L1 Norm Diff . . . . .                                 | 2536 |
| 7.195.1 Function Documentation . . . . .                    | 2537 |
| 7.195.1.1 nppsNormDiff_L1_16s32f . . . . .                  | 2537 |
| 7.195.1.2 nppsNormDiff_L1_16s32s_Sfs . . . . .              | 2537 |
| 7.195.1.3 nppsNormDiff_L1_16s64s_Sfs . . . . .              | 2537 |
| 7.195.1.4 nppsNormDiff_L1_32f . . . . .                     | 2538 |
| 7.195.1.5 nppsNormDiff_L1_32fc64f . . . . .                 | 2538 |
| 7.195.1.6 nppsNormDiff_L1_64f . . . . .                     | 2539 |
| 7.195.1.7 nppsNormDiff_L1_64fc64f . . . . .                 | 2539 |
| 7.195.1.8 nppsNormDiffL1GetBufferSize_16s32f . . . . .      | 2539 |
| 7.195.1.9 nppsNormDiffL1GetBufferSize_16s32s_Sfs . . . . .  | 2540 |
| 7.195.1.10 nppsNormDiffL1GetBufferSize_16s64s_Sfs . . . . . | 2540 |
| 7.195.1.11 nppsNormDiffL1GetBufferSize_32f . . . . .        | 2540 |
| 7.195.1.12 nppsNormDiffL1GetBufferSize_32fc64f . . . . .    | 2540 |
| 7.195.1.13 nppsNormDiffL1GetBufferSize_64f . . . . .        | 2541 |
| 7.195.1.14 nppsNormDiffL1GetBufferSize_64fc64f . . . . .    | 2541 |
| 7.196L2 Norm Diff . . . . .                                 | 2542 |
| 7.196.1 Function Documentation . . . . .                    | 2543 |
| 7.196.1.1 nppsNormDiff_L2_16s32f . . . . .                  | 2543 |
| 7.196.1.2 nppsNormDiff_L2_16s32s_Sfs . . . . .              | 2543 |
| 7.196.1.3 nppsNormDiff_L2_32f . . . . .                     | 2543 |
| 7.196.1.4 nppsNormDiff_L2_32fc64f . . . . .                 | 2544 |
| 7.196.1.5 nppsNormDiff_L2_64f . . . . .                     | 2544 |
| 7.196.1.6 nppsNormDiff_L2_64fc64f . . . . .                 | 2545 |
| 7.196.1.7 nppsNormDiff_L2Sqr_16s64s_Sfs . . . . .           | 2545 |
| 7.196.1.8 nppsNormDiffL2GetBufferSize_16s32f . . . . .      | 2545 |

|  |      |
|--|------|
| 7.196.1.9 nppsNormDiffL2GetBufferSize_16s32s_Sfs . . . . .     | 2546 |
| 7.196.1.10 nppsNormDiffL2GetBufferSize_32f . . . . .           | 2546 |
| 7.196.1.11 nppsNormDiffL2GetBufferSize_32fc64f . . . . .       | 2546 |
| 7.196.1.12 nppsNormDiffL2GetBufferSize_64f . . . . .           | 2546 |
| 7.196.1.13 nppsNormDiffL2GetBufferSize_64fc64f . . . . .       | 2547 |
| 7.196.1.14 nppsNormDiffL2SqrGetBufferSize_16s64s_Sfs . . . . . | 2547 |
| 7.197 Dot Product . . . . .                                    | 2548 |
| 7.197.1 Function Documentation . . . . .                       | 2551 |
| 7.197.1.1 nppsDotProd_16s16sc32fc . . . . .                    | 2551 |
| 7.197.1.2 nppsDotProd_16s16sc32sc_Sfs . . . . .                | 2552 |
| 7.197.1.3 nppsDotProd_16s16sc64sc . . . . .                    | 2552 |
| 7.197.1.4 nppsDotProd_16s16sc_Sfs . . . . .                    | 2553 |
| 7.197.1.5 nppsDotProd_16s32f . . . . .                         | 2553 |
| 7.197.1.6 nppsDotProd_16s32s32s_Sfs . . . . .                  | 2553 |
| 7.197.1.7 nppsDotProd_16s32s_Sfs . . . . .                     | 2554 |
| 7.197.1.8 nppsDotProd_16s64s . . . . .                         | 2554 |
| 7.197.1.9 nppsDotProd_16s_Sfs . . . . .                        | 2555 |
| 7.197.1.10 nppsDotProd_16sc32fc . . . . .                      | 2555 |
| 7.197.1.11 nppsDotProd_16sc32sc_Sfs . . . . .                  | 2555 |
| 7.197.1.12 nppsDotProd_16sc64sc . . . . .                      | 2556 |
| 7.197.1.13 nppsDotProd_16sc_Sfs . . . . .                      | 2556 |
| 7.197.1.14 nppsDotProd_32f . . . . .                           | 2557 |
| 7.197.1.15 nppsDotProd_32f32fc . . . . .                       | 2557 |
| 7.197.1.16 nppsDotProd_32f32fc64fc . . . . .                   | 2557 |
| 7.197.1.17 nppsDotProd_32f64f . . . . .                        | 2558 |
| 7.197.1.18 nppsDotProd_32fc . . . . .                          | 2558 |
| 7.197.1.19 nppsDotProd_32fc64fc . . . . .                      | 2558 |
| 7.197.1.20 nppsDotProd_32s32sc_Sfs . . . . .                   | 2559 |
| 7.197.1.21 nppsDotProd_32s_Sfs . . . . .                       | 2559 |
| 7.197.1.22 nppsDotProd_32sc_Sfs . . . . .                      | 2559 |
| 7.197.1.23 nppsDotProd_64f . . . . .                           | 2560 |
| 7.197.1.24 nppsDotProd_64f64fc . . . . .                       | 2560 |
| 7.197.1.25 nppsDotProd_64fc . . . . .                          | 2561 |
| 7.197.1.26 nppsDotProdGetBufferSize_16s16sc32fc . . . . .      | 2561 |
| 7.197.1.27 nppsDotProdGetBufferSize_16s16sc32sc_Sfs . . . . .  | 2561 |
| 7.197.1.28 nppsDotProdGetBufferSize_16s16sc64sc . . . . .      | 2561 |

|            |  |      |
|------------|--|------|
| 7.197.1.29 | nppsDotProdGetBufferSize_16s16sc_Sfs . . . . .   | 2562 |
| 7.197.1.30 | nppsDotProdGetBufferSize_16s32f . . . . .        | 2562 |
| 7.197.1.31 | nppsDotProdGetBufferSize_16s32s32s_Sfs . . . . . | 2562 |
| 7.197.1.32 | nppsDotProdGetBufferSize_16s32s_Sfs . . . . .    | 2563 |
| 7.197.1.33 | nppsDotProdGetBufferSize_16s64s . . . . .        | 2563 |
| 7.197.1.34 | nppsDotProdGetBufferSize_16s_Sfs . . . . .       | 2563 |
| 7.197.1.35 | nppsDotProdGetBufferSize_16sc32fc . . . . .      | 2563 |
| 7.197.1.36 | nppsDotProdGetBufferSize_16sc32sc_Sfs . . . . .  | 2564 |
| 7.197.1.37 | nppsDotProdGetBufferSize_16sc64sc . . . . .      | 2564 |
| 7.197.1.38 | nppsDotProdGetBufferSize_16sc_Sfs . . . . .      | 2564 |
| 7.197.1.39 | nppsDotProdGetBufferSize_32f . . . . .           | 2564 |
| 7.197.1.40 | nppsDotProdGetBufferSize_32f32fc . . . . .       | 2565 |
| 7.197.1.41 | nppsDotProdGetBufferSize_32f32fc64fc . . . . .   | 2565 |
| 7.197.1.42 | nppsDotProdGetBufferSize_32f64f . . . . .        | 2565 |
| 7.197.1.43 | nppsDotProdGetBufferSize_32fc . . . . .          | 2565 |
| 7.197.1.44 | nppsDotProdGetBufferSize_32fc64fc . . . . .      | 2566 |
| 7.197.1.45 | nppsDotProdGetBufferSize_32s32sc_Sfs . . . . .   | 2566 |
| 7.197.1.46 | nppsDotProdGetBufferSize_32s_Sfs . . . . .       | 2566 |
| 7.197.1.47 | nppsDotProdGetBufferSize_32sc_Sfs . . . . .      | 2566 |
| 7.197.1.48 | nppsDotProdGetBufferSize_64f . . . . .           | 2567 |
| 7.197.1.49 | nppsDotProdGetBufferSize_64f64fc . . . . .       | 2567 |
| 7.197.1.50 | nppsDotProdGetBufferSize_64fc . . . . .          | 2567 |
| 7.198      | Count In Range . . . . .                         | 2568 |
| 7.198.1    | Function Documentation . . . . .                 | 2568 |
| 7.198.1.1  | nppsCountInRange_32s . . . . .                   | 2568 |
| 7.198.1.2  | nppsCountInRangeGetBufferSize_32s . . . . .      | 2568 |
| 7.199      | Count Zero Crossings . . . . .                   | 2569 |
| 7.199.1    | Function Documentation . . . . .                 | 2569 |
| 7.199.1.1  | nppsZeroCrossing_16s32f . . . . .                | 2569 |
| 7.199.1.2  | nppsZeroCrossing_32f . . . . .                   | 2569 |
| 7.199.1.3  | nppsZeroCrossingGetBufferSize_16s32f . . . . .   | 2570 |
| 7.199.1.4  | nppsZeroCrossingGetBufferSize_32f . . . . .      | 2570 |
| 7.200      | MaximumError . . . . .                           | 2571 |
| 7.200.1    | Detailed Description . . . . .                   | 2573 |
| 7.200.2    | Function Documentation . . . . .                 | 2573 |
| 7.200.2.1  | nppsMaximumError_16s . . . . .                   | 2573 |

|   |      |
|---|------|
| 7.200.2.2 nppsMaximumError_16s                | 2573 |
| 7.200.2.3 nppsMaximumError_16u                | 2574 |
| 7.200.2.4 nppsMaximumError_32f                | 2574 |
| 7.200.2.5 nppsMaximumError_32fc               | 2574 |
| 7.200.2.6 nppsMaximumError_32s                | 2575 |
| 7.200.2.7 nppsMaximumError_32sc               | 2575 |
| 7.200.2.8 nppsMaximumError_32u                | 2575 |
| 7.200.2.9 nppsMaximumError_64f                | 2576 |
| 7.200.2.10 nppsMaximumError_64fc              | 2576 |
| 7.200.2.11 nppsMaximumError_64s               | 2576 |
| 7.200.2.12 nppsMaximumError_64sc              | 2577 |
| 7.200.2.13 nppsMaximumError_8s                | 2577 |
| 7.200.2.14 nppsMaximumError_8u                | 2577 |
| 7.200.2.15 nppsMaximumErrorGetBufferSize_16s  | 2578 |
| 7.200.2.16 nppsMaximumErrorGetBufferSize_16sc | 2578 |
| 7.200.2.17 nppsMaximumErrorGetBufferSize_16u  | 2578 |
| 7.200.2.18 nppsMaximumErrorGetBufferSize_32f  | 2578 |
| 7.200.2.19 nppsMaximumErrorGetBufferSize_32fc | 2579 |
| 7.200.2.20 nppsMaximumErrorGetBufferSize_32s  | 2579 |
| 7.200.2.21 nppsMaximumErrorGetBufferSize_32sc | 2579 |
| 7.200.2.22 nppsMaximumErrorGetBufferSize_32u  | 2579 |
| 7.200.2.23 nppsMaximumErrorGetBufferSize_64f  | 2580 |
| 7.200.2.24 nppsMaximumErrorGetBufferSize_64fc | 2580 |
| 7.200.2.25 nppsMaximumErrorGetBufferSize_64s  | 2580 |
| 7.200.2.26 nppsMaximumErrorGetBufferSize_64sc | 2580 |
| 7.200.2.27 nppsMaximumErrorGetBufferSize_8s   | 2581 |
| 7.200.2.28 nppsMaximumErrorGetBufferSize_8u   | 2581 |
| 7.201 AverageError                            | 2582 |
| 7.201.1 Detailed Description                  | 2584 |
| 7.201.2 Function Documentation                | 2584 |
| 7.201.2.1 nppsAverageError_16s                | 2584 |
| 7.201.2.2 nppsAverageError_16sc               | 2584 |
| 7.201.2.3 nppsAverageError_16u                | 2585 |
| 7.201.2.4 nppsAverageError_32f                | 2585 |
| 7.201.2.5 nppsAverageError_32fc               | 2585 |
| 7.201.2.6 nppsAverageError_32s                | 2586 |

|   |      |
|---|------|
| 7.201.2.7 nppsAverageError_32sc . . . . .               | 2586 |
| 7.201.2.8 nppsAverageError_32u . . . . .                | 2586 |
| 7.201.2.9 nppsAverageError_64f . . . . .                | 2587 |
| 7.201.2.10 nppsAverageError_64fc . . . . .              | 2587 |
| 7.201.2.11 nppsAverageError_64s . . . . .               | 2587 |
| 7.201.2.12 nppsAverageError_64sc . . . . .              | 2588 |
| 7.201.2.13 nppsAverageError_8s . . . . .                | 2588 |
| 7.201.2.14 nppsAverageError_8u . . . . .                | 2588 |
| 7.201.2.15 nppsAverageErrorGetBufferSize_16s . . . . .  | 2589 |
| 7.201.2.16 nppsAverageErrorGetBufferSize_16sc . . . . . | 2589 |
| 7.201.2.17 nppsAverageErrorGetBufferSize_16u . . . . .  | 2589 |
| 7.201.2.18 nppsAverageErrorGetBufferSize_32f . . . . .  | 2589 |
| 7.201.2.19 nppsAverageErrorGetBufferSize_32fc . . . . . | 2590 |
| 7.201.2.20 nppsAverageErrorGetBufferSize_32s . . . . .  | 2590 |
| 7.201.2.21 nppsAverageErrorGetBufferSize_32sc . . . . . | 2590 |
| 7.201.2.22 nppsAverageErrorGetBufferSize_32u . . . . .  | 2590 |
| 7.201.2.23 nppsAverageErrorGetBufferSize_64f . . . . .  | 2591 |
| 7.201.2.24 nppsAverageErrorGetBufferSize_64fc . . . . . | 2591 |
| 7.201.2.25 nppsAverageErrorGetBufferSize_64s . . . . .  | 2591 |
| 7.201.2.26 nppsAverageErrorGetBufferSize_64sc . . . . . | 2591 |
| 7.201.2.27 nppsAverageErrorGetBufferSize_8s . . . . .   | 2592 |
| 7.201.2.28 nppsAverageErrorGetBufferSize_8u . . . . .   | 2592 |
| 7.202 MaximumRelativeError . . . . .                    | 2593 |
| 7.202.1 Detailed Description . . . . .                  | 2595 |
| 7.202.2 Function Documentation . . . . .                | 2595 |
| 7.202.2.1 nppsMaximumRelativeError_16s . . . . .        | 2595 |
| 7.202.2.2 nppsMaximumRelativeError_16sc . . . . .       | 2595 |
| 7.202.2.3 nppsMaximumRelativeError_16u . . . . .        | 2596 |
| 7.202.2.4 nppsMaximumRelativeError_32f . . . . .        | 2596 |
| 7.202.2.5 nppsMaximumRelativeError_32fc . . . . .       | 2597 |
| 7.202.2.6 nppsMaximumRelativeError_32s . . . . .        | 2597 |
| 7.202.2.7 nppsMaximumRelativeError_32sc . . . . .       | 2597 |
| 7.202.2.8 nppsMaximumRelativeError_32u . . . . .        | 2598 |
| 7.202.2.9 nppsMaximumRelativeError_64f . . . . .        | 2598 |
| 7.202.2.10 nppsMaximumRelativeError_64fc . . . . .      | 2599 |
| 7.202.2.11 nppsMaximumRelativeError_64s . . . . .       | 2599 |



|  |      |
|--|------|
| 7.202.2.12nppsMaximumRelativeError_64sc . . . . .              | 2599 |
| 7.202.2.13nppsMaximumRelativeError_8s . . . . .                | 2600 |
| 7.202.2.14nppsMaximumRelativeError_8u . . . . .                | 2600 |
| 7.202.2.15nppsMaximumRelativeErrorGetBufferSize_16s . . . . .  | 2601 |
| 7.202.2.16nppsMaximumRelativeErrorGetBufferSize_16sc . . . . . | 2601 |
| 7.202.2.17nppsMaximumRelativeErrorGetBufferSize_16u . . . . .  | 2601 |
| 7.202.2.18nppsMaximumRelativeErrorGetBufferSize_32f . . . . .  | 2601 |
| 7.202.2.19nppsMaximumRelativeErrorGetBufferSize_32fc . . . . . | 2602 |
| 7.202.2.20nppsMaximumRelativeErrorGetBufferSize_32s . . . . .  | 2602 |
| 7.202.2.21nppsMaximumRelativeErrorGetBufferSize_32sc . . . . . | 2602 |
| 7.202.2.22nppsMaximumRelativeErrorGetBufferSize_32u . . . . .  | 2602 |
| 7.202.2.23nppsMaximumRelativeErrorGetBufferSize_64f . . . . .  | 2603 |
| 7.202.2.24nppsMaximumRelativeErrorGetBufferSize_64fc . . . . . | 2603 |
| 7.202.2.25nppsMaximumRelativeErrorGetBufferSize_64s . . . . .  | 2603 |
| 7.202.2.26nppsMaximumRelativeErrorGetBufferSize_64sc . . . . . | 2603 |
| 7.202.2.27nppsMaximumRelativeErrorGetBufferSize_8s . . . . .   | 2604 |
| 7.202.2.28nppsMaximumRelativeErrorGetBufferSize_8u . . . . .   | 2604 |
| 7.203 AverageRelativeError . . . . .                           | 2605 |
| 7.203.1 Detailed Description . . . . .                         | 2607 |
| 7.203.2 Function Documentation . . . . .                       | 2607 |
| 7.203.2.1 nppsAverageRelativeError_16s . . . . .               | 2607 |
| 7.203.2.2 nppsAverageRelativeError_16sc . . . . .              | 2607 |
| 7.203.2.3 nppsAverageRelativeError_16u . . . . .               | 2608 |
| 7.203.2.4 nppsAverageRelativeError_32f . . . . .               | 2608 |
| 7.203.2.5 nppsAverageRelativeError_32fc . . . . .              | 2609 |
| 7.203.2.6 nppsAverageRelativeError_32s . . . . .               | 2609 |
| 7.203.2.7 nppsAverageRelativeError_32sc . . . . .              | 2609 |
| 7.203.2.8 nppsAverageRelativeError_32u . . . . .               | 2610 |
| 7.203.2.9 nppsAverageRelativeError_64f . . . . .               | 2610 |
| 7.203.2.10nppsAverageRelativeError_64fc . . . . .              | 2611 |
| 7.203.2.11nppsAverageRelativeError_64s . . . . .               | 2611 |
| 7.203.2.12nppsAverageRelativeError_64sc . . . . .              | 2611 |
| 7.203.2.13nppsAverageRelativeError_8s . . . . .                | 2612 |
| 7.203.2.14nppsAverageRelativeError_8u . . . . .                | 2612 |
| 7.203.2.15nppsAverageRelativeErrorGetBufferSize_16s . . . . .  | 2613 |
| 7.203.2.16nppsAverageRelativeErrorGetBufferSize_16sc . . . . . | 2613 |

|            |  |             |
|------------|--|-------------|
| 7.203.2.17 | nppsAverageRelativeErrorGetBufferSize_16u  | 2613        |
| 7.203.2.18 | nppsAverageRelativeErrorGetBufferSize_32f  | 2613        |
| 7.203.2.19 | nppsAverageRelativeErrorGetBufferSize_32fc | 2614        |
| 7.203.2.20 | nppsAverageRelativeErrorGetBufferSize_32s  | 2614        |
| 7.203.2.21 | nppsAverageRelativeErrorGetBufferSize_32sc | 2614        |
| 7.203.2.22 | nppsAverageRelativeErrorGetBufferSize_32u  | 2614        |
| 7.203.2.23 | nppsAverageRelativeErrorGetBufferSize_64f  | 2615        |
| 7.203.2.24 | nppsAverageRelativeErrorGetBufferSize_64fc | 2615        |
| 7.203.2.25 | nppsAverageRelativeErrorGetBufferSize_64s  | 2615        |
| 7.203.2.26 | nppsAverageRelativeErrorGetBufferSize_64sc | 2615        |
| 7.203.2.27 | nppsAverageRelativeErrorGetBufferSize_8s   | 2616        |
| 7.203.2.28 | nppsAverageRelativeErrorGetBufferSize_8u   | 2616        |
| 7.204      | Memory Management                          | 2617        |
| 7.205      | Malloc                                     | 2618        |
| 7.205.1    | Detailed Description                       | 2619        |
| 7.205.2    | Function Documentation                     | 2619        |
| 7.205.2.1  | nppsMalloc_16s                             | 2619        |
| 7.205.2.2  | nppsMalloc_16sc                            | 2619        |
| 7.205.2.3  | nppsMalloc_16u                             | 2619        |
| 7.205.2.4  | nppsMalloc_32f                             | 2620        |
| 7.205.2.5  | nppsMalloc_32fc                            | 2620        |
| 7.205.2.6  | nppsMalloc_32s                             | 2620        |
| 7.205.2.7  | nppsMalloc_32sc                            | 2620        |
| 7.205.2.8  | nppsMalloc_32u                             | 2621        |
| 7.205.2.9  | nppsMalloc_64f                             | 2621        |
| 7.205.2.10 | nppsMalloc_64fc                            | 2621        |
| 7.205.2.11 | nppsMalloc_64s                             | 2621        |
| 7.205.2.12 | nppsMalloc_64sc                            | 2622        |
| 7.205.2.13 | nppsMalloc_8s                              | 2622        |
| 7.205.2.14 | nppsMalloc_8u                              | 2622        |
| 7.206      | Free                                       | 2623        |
| 7.206.1    | Detailed Description                       | 2623        |
| 7.206.2    | Function Documentation                     | 2623        |
| 7.206.2.1  | nppsFree                                   | 2623        |
| <b>8</b>   | <b>Data Structure Documentation</b>        | <b>2625</b> |
| 8.1        | NPP_ALIGN_16 Struct Reference              | 2625        |

|         |   |      |
|---------|---|------|
| 8.1.1   | Detailed Description                    | 2625 |
| 8.1.2   | Field Documentation                     | 2625 |
| 8.1.2.1 | im                                      | 2625 |
| 8.1.2.2 | im                                      | 2626 |
| 8.1.2.3 | re                                      | 2626 |
| 8.1.2.4 | re                                      | 2626 |
| 8.2     | NPP_ALIGN_8 Struct Reference            | 2627 |
| 8.2.1   | Detailed Description                    | 2627 |
| 8.2.2   | Field Documentation                     | 2627 |
| 8.2.2.1 | im                                      | 2627 |
| 8.2.2.2 | im                                      | 2627 |
| 8.2.2.3 | im                                      | 2627 |
| 8.2.2.4 | re                                      | 2628 |
| 8.2.2.5 | re                                      | 2628 |
| 8.2.2.6 | re                                      | 2628 |
| 8.3     | NppiHaarBuffer Struct Reference         | 2629 |
| 8.3.1   | Field Documentation                     | 2629 |
| 8.3.1.1 | haarBuffer                              | 2629 |
| 8.3.1.2 | haarBufferSize                          | 2629 |
| 8.4     | NppiHaarClassifier_32f Struct Reference | 2630 |
| 8.4.1   | Field Documentation                     | 2630 |
| 8.4.1.1 | classifiers                             | 2630 |
| 8.4.1.2 | classifierSize                          | 2630 |
| 8.4.1.3 | classifierStep                          | 2630 |
| 8.4.1.4 | counterDevice                           | 2630 |
| 8.4.1.5 | numClassifiers                          | 2630 |
| 8.5     | NppiPoint Struct Reference              | 2631 |
| 8.5.1   | Detailed Description                    | 2631 |
| 8.5.2   | Field Documentation                     | 2631 |
| 8.5.2.1 | x                                       | 2631 |
| 8.5.2.2 | y                                       | 2631 |
| 8.6     | NppiRect Struct Reference               | 2632 |
| 8.6.1   | Detailed Description                    | 2632 |
| 8.6.2   | Field Documentation                     | 2632 |
| 8.6.2.1 | height                                  | 2632 |
| 8.6.2.2 | width                                   | 2632 |

|         |                                    |      |
|---------|------------------------------------|------|
| 8.6.2.3 | x                                  | 2632 |
| 8.6.2.4 | y                                  | 2632 |
| 8.7     | NppiSize Struct Reference          | 2633 |
| 8.7.1   | Detailed Description               | 2633 |
| 8.7.2   | Field Documentation                | 2633 |
| 8.7.2.1 | height                             | 2633 |
| 8.7.2.2 | width                              | 2633 |
| 8.8     | NppLibraryVersion Struct Reference | 2634 |
| 8.8.1   | Field Documentation                | 2634 |
| 8.8.1.1 | build                              | 2634 |
| 8.8.1.2 | major                              | 2634 |
| 8.8.1.3 | minor                              | 2634 |

# Chapter 1

## NVIDIA Performance Primitives

IMPORTANT SPECIAL NOTICE IMPORTANT SPECIAL NOTICE IMPORTANT SPECIAL NOTICE  
As of NPP version 5.0 and beyond a few parameters for a few pre-5.0 existing image LUT functions have changed from host memory pointers to device memory pointers. Your application will fail (crash or report an error) if you use these functions with host memory pointers. The functions are the `nppiLUT_Linear_8u_xxx` functions.

Also, pre-5.0 function `nppiMeanStdDev8uC1RGetBufferHostSize` has been renamed `nppiMeanStdDevGetBufferHostSize_8u_C1R`.

### 1.1 What is NPP?

NVIDIA NPP is a library of functions for performing CUDA accelerated processing. The initial set of functionality in the library focuses on imaging and video processing and is widely applicable for developers in these areas. NPP will evolve over time to encompass more of the compute heavy tasks in a variety of problem domains. The NPP library is written to maximize flexibility, while maintaining high performance.

NPP can be used in one of two ways:

- A stand-alone library for adding GPU acceleration to an application with minimal effort. Using this route allows developers to add GPU acceleration to their applications in a matter of hours.
- A cooperative library for interoperating with a developer's GPU code efficiently.

Either route allows developers to harness the massive compute resources of NVIDIA GPUs, while simultaneously reducing development times.

### 1.2 Documentation

- [General API Conventions](#)
- [Signal-Processing Specific API Conventions](#)
- [Imaging-Processing Specific API Conventions](#)

## 1.3 Technical Specifications

Supported Platforms:

- Microsoft Windows 7 and 8 (64-bit and 32-bit)
- Microsoft Windows Vista (64-bit and 32-bit)
- Linux (Centos & Ubuntu) (64-bit and 32-bit)
- Mac OS X (64-bit)
- Android on Arm V7

## 1.4 Files

NPP is comprises the following files:

### 1.4.1 Header Files

- [nppdefs.h](#)
- [nppcore.h](#)
- [nppi.h](#)
- [npps.h](#)
- [nppversion.h](#)
- [npp.h](#)

All those header files are located in the CUDA Toolkit's

`/include/`

directory.

### 1.4.2 Library Files

Starting with Version 5.5 NPP's functionality is now split up into 3 distinct libraries:

- A core library (NPPC) containing basic functionality from the [npp.h](#) header files as well as functionality shared by the other two libraries.
- The image processing library NPPI. Any functions from the [nppi.h](#) header file (or the various header files named "nppi\_xxx.h" are bundled into the NPPI library.
- The signal processing library NPPS. Any function from the [npps.h](#) header file (or the various header files named "npps\_xxx.h" are bundled into the NPPS library.

On the Windows platform the NPP stub libraries are found in the CUDA Toolkit's library directory:

```
/lib/nppc.lib
```

```
/lib/nppi.lib
```

```
/lib/npps.lib
```

The matching DLLs are located in the CUDA Toolkit's binary directory. Example

```
/bin/nppi64_55_<build_no>.dll      // Dynamic image-processing library for 64-bit Windows.
```

On Linux and Mac platforms the dynamic libraries are located in the lib directory

```
/lib/libnppc32.so.5.5.<build_no>    // NPP 32-bit dynamic core library for Linux
```

```
/lib/libnpps32.5.5.dylib           // NPP 32-bit dynamic signal processing library for Mac
```

## 1.5 Supported NVIDIA Hardware

NPP runs on all CUDA capable NVIDIA hardware. For details please see [http://www.nvidia.com/object/cuda\\_learn\\_products.html](http://www.nvidia.com/object/cuda_learn_products.html)





## **Chapter 2**

# **General API Conventions**

## 2.1 Memory Management

The design of all the NPP functions follows the same guidelines as other NVIDIA CUDA libraries like cuFFT and cuBLAS. That is that all pointer arguments in those APIs are device pointers.

This convention enables the individual developer to make smart choices about memory management that minimize the number of memory transfers. It also allows the user the maximum flexibility regarding which of the various memory transfer mechanisms offered by the CUDA runtime is used, e.g. synchronous or asynchronous memory transfers, zero-copy and pinned memory, etc.

The most basic steps involved in using NPP for processing data is as follows:

1. Transfer input data from the host to device using

```
cudaMemcpy(...)
```

2. Process data using one or several NPP functions or custom CUDA kernels

3. Transfer the result data from the device to the host using

```
cudaMemcpy(...)
```

### 2.1.1 Scratch Buffer and Host Pointer

Some primitives of NPP require additional device memory buffers (scratch buffers) for calculations, e.g. signal and image reductions (Sum, Max, Min, MinMax, etc.). In order to give the NPP user maximum control regarding memory allocations and performance, it is the user's responsibility to allocate and delete those temporary buffers. For one this has the benefit that the library will not allocate memory unbeknownst to the user. It also allows developers who invoke the same primitive repeatedly to allocate the scratch only once, improving performance and potential device-memory fragmentation.

Scratch-buffer memory is unstructured and may be passed to the primitive in uninitialized form. This allows for reuse of the same scratch buffers with any primitive require scratch memory, as long as it is sufficiently sized.

The minimum scratch-buffer size for a given primitive (e.g. `nppsSum_32f()`) can be obtained by a companion function (e.g. `nppsSumGetBufferSize_32f()`). The buffer size is returned via a host pointer as allocation of the scratch-buffer is performed via CUDA runtime host code.

An example to invoke signal sum primitive and allocate and free the necessary scratch memory:

```
// pSrc, pSum, pDeviceBuffer are all device pointers.
Npp32f * pSrc;
Npp32f * pSum;
Npp8u * pDeviceBuffer;
int nLength = 1024;

// Allocate the device memroy.
cudaMalloc((void **)&pSrc, sizeof(Npp32f) * nLength);
nppsSet_32f(1.0f, pSrc, nLength);
cudaMalloc((void **)&pSum, sizeof(Npp32f) * 1);

// Compute the appropriate size of the scratch-memory buffer
int nBufferSize;
nppsSumGetBufferSize_32f(nLength, &nBufferSize);
// Allocate the scratch buffer
cudaMalloc((void **)&pDeviceBuffer, nBufferSize);

// Call the primitive with the scratch buffer
```

```

nppsSum_32f(pSrc, nLength, pSum, pDeviceBuffer);
Npp32f nSumHost;
cudaMemcpy(&nSumHost, pSum, sizeof(Npp32f) * 1, cudaMemcpyDeviceToHost);
printf("sum = %f\n", nSumHost); // nSumHost = 1024.0f;

// Free the device memory
cudaFree(pSrc);
cudaFree(pDeviceBuffer);
cudaFree(pSum);

```

## 2.2 Function Naming

Since NPP is a C API and therefore does not allow for function overloading for different data-types the NPP naming convention addresses the need to differentiate between different flavors of the same algorithm or primitive function but for various data types. This disambiguation of different flavors of a primitive is done via a suffix containing data type and other disambiguating information.

In addition to the flavor suffix, all NPP functions are prefixed with by the letters "npp". Primitives belonging to NPP's image-processing module add the letter "i" to the npp prefix, i.e. are prefixed by "nppi". Similarly signal-processing primitives are prefixed with "npps".

The general naming scheme is:

npp<module info><PrimitiveName>\_<data-type info>[\_<additional flavor info>](<parameter list>)

The data-type information uses the same names as the [Basic NPP Data Types](#). For example the data-type information "8u" would imply that the primitive operates on [Npp8u](#) data.

If a primitive consumes different type data from what it produces, both types will be listed in the order of consumed to produced data type.

Details about the "additional flavor information" is provided for each of the NPP modules, since each problem domain uses different flavor information suffixes.

## 2.3 Integer Result Scaling

NPP signal processing and imaging primitives often operate on integer data. This integer data is usually a fixed point fractional representation of some physical magnitue (e.g. luminance). Because of this fixed-point nature of the representation many numerical operations (e.g. addition or multiplication) tend to produce results exceeding the original fixed-point range if treated as regular integers.

In cases where the results exceed the original range, these functions clamp the result values back to the valid range. E.g. the maximum positive value for a 16-bit unsigned integer is 32767. A multiplication operation of  $4 * 10000 = 40000$  would exceed this range. The result would be clamped to be 32767.

To avoid the level of lost information due to clamping most integer primitives allow for result scaling. Primitives with result scaling have the "Sfs" suffix in their name and provide a parameter "nScaleFactor" that controls the amount of scaling. Before the results of an operation are clamped to the valid output-data range by multiplying them with  $2^{-nScaleFactor}$ .

Example: The primitive [nppsSqr\\_8u\\_Sfs\(\)](#) computes the square of 8-bit unsigned sample values in a signal (1D array of values). The maximum value of a 8-bit value is 255. The square of  $255^2 = 65025$  which would be clamped to 255 if no result scaling is performed. In order to map the maximum value of 255 to 255 in the result, one would specify an integer result scaling factor of 8, i.e. multiply each result with  $2^{-8} = \frac{1}{2^8} = \frac{1}{256}$ . The final result for a signal value of 255 being squared and scaled would be:

$$255^2 \cdot 2^{-8} = 254.00390625$$

which would be rounded to a final result of 254.

A medium gray value of 128 would result in

$$128^2 * 2^{-8} = 64$$

## 2.4 Rounding Modes

Many NPP functions require converting floating-point values to integers. The [NppRoundMode](#) enum lists NPP's supported rounding modes. Not all primitives in NPP that perform rounding as part of their functionality allow the user to specify the round-mode used. Instead they use NPP's default rounding mode, which is [NPP\\_RND\\_FINANCIAL](#).

### 2.4.1 Rounding Mode Parameter

A subset of NPP functions performing rounding as part of their functionality do allow the user to specify which rounding mode is used through a parameter of the [NppRoundMode](#) type.

## **Chapter 3**

# **Signal-Processing Specific API Conventions**

## 3.1 Signal Data

Signal data is passed to and from NPPS primitives via a pointer to the signal's data type.

The general idea behind this fairly low-level way of passing signal data is ease-of-adoption into existing software projects:

- Passing the data pointer rather than a higher-level signal struct allows for easy adoption by not requiring a specific signal representation (that could include total signal size offset, or other additional information). This avoids awkward packing and unpacking of signal data from the host application to an NPP specific signal representation.

### 3.1.1 Parameter Names for Signal Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

Those are signals consumed by the algorithm.

#### 3.1.1.1 Source Signal Pointer

The source signal data is generally passed via a pointer named

`pSrc`

The source signal pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
nppsPrimitive_32s(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

`pSrc1, pSrc2, ...`

#### 3.1.1.2 Destination Signal Pointer

The destination signal data is generally passed via a pointer named

`pDst`

In case the primitive consumes multiple signals as inputs the source pointers are numbered like this:

`pDst1, pDst2, ...`

#### 3.1.1.3 In-Place Signal Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place signal data are called:

`pSrcDst`

### 3.1.2 Signal Data Alignment Requirements

NPP requires signal sample data to be naturally aligned, i.e. any pointer

```
NppType * p;
```

to a sample in a signal needs to fulfill:

```
assert(p % sizeof(p) == 0);
```

### 3.1.3 Signal Data Related Error Codes

All NPPI primitives operating on signal data validate the signal-data pointer for proper alignment and test that the point is not null.

Failed validation results in one of the following error codes being returned and the primitive not being executed:

- [NPP\\_NULL\\_POINTER\\_ERROR](#) is returned if the image-data pointer is 0 (NULL).
- [NPP\\_ALIGNMENT\\_ERROR](#) if the signal-data pointer address is not a multiple of the signal's data-type size.

## 3.2 Signal Length

The vast majority of NPPS functions take a

```
nLength
```

parameter that tells the primitive how many of the signal's samples starting from the given data pointer are to be processed.

### 3.2.1 Length Related Error Codes

All NPPS primitives taking a length parameter validate this input.

Failed validation results in the following error code being returned and the primitive not being executed:

- [NPP\\_SIZE\\_ERROR](#) is returned if the length is negative.





## **Chapter 4**

# **Imaging-Processing Specific API Conventions**

## 4.1 Function Naming

Image processing related functions use a number of suffixes to indicate various different flavors of a primitive beyond just different data types. The flavor suffix uses the following abbreviations:

- "A" if the image is a 4 channel image this indicates the result alpha channel is not affected by the primitive.
- "Cn" the image consists of n channel packed pixels, where n can be 1, 2, 3 or 4.
- "Pn" the image consists of n separate image planes, where n can be 1, 2, 3 or 4.
- "C" (following the channel information) indicates that the primitive only operates on one of the color channels, the "channel-of-interest". All other output channels are not affected by the primitive.
- "I" indicates that the primitive works "in-place". In this case the image-data pointer is usually named "pSrcDst" to indicate that the image data serves as source and destination at the same time.
- "M" indicates "masked operation". These types of primitives have an additional "mask image" as input. Each pixel in the destination image corresponds to a pixel in the mask image. Only pixels with a corresponding non-zero mask pixel are being processed.
- "R" indicates the primitive operates only on a rectangular "region-of-interest" or "ROI". All ROI primitives take an additional input parameter of type [NppiSize](#), which specifies the width and height of the rectangular region that the primitive should process. For details on how primitives operate on ROIs see: [Region-of-Interest \(ROI\)](#).
- "Sfs" indicates the result values are processed by fixed scaling and saturation before they're written out.

The suffixes above always appear in alphabetical order. E.g. a 4 channel primitive not affecting the alpha channel with masked operation, in place and with scaling/saturation and ROI would have the postfix: "AC4IMRSfs".

## 4.2 Image Data

Image data is passed to and from NPPI primitives via a pair of parameters:

1. A pointer to the image's underlying data type.
2. A line step in bytes (also sometimes called line stride).

The general idea behind this fairly low-level way of passing image data is ease-of-adoption into existing software projects:

- Passing a raw pointer to the underlying pixel data type, rather than structured (by color) channel pixel data allows usage of the function in a wide variety of situations avoiding risky type cast or expensive image data copies.
- Passing the data pointer and line step individually rather than a higher-level image struct again allows for easy adoption by not requiring a specific image representation and thus avoiding awkward packing and unpacking of image data from the host application to an NPP specific image representation.

### 4.2.1 Line Step

The line step (also called "line stride" or "row step") allows lines of oddly sized images to start on well-aligned addresses by adding a number of unused bytes at the ends of the lines. This type of line padding has been common practice in digital image processing for a long time and is not particular to GPU image processing.

The line step is the number of bytes in a line **including the padding**. An other way to interpret this number is to say that it is the number of bytes between the first pixel of successive rows in the image, or generally the number of bytes between two neighboring pixels in any column of pixels.

The general reason for the existence of the line step it is that uniformly aligned rows of pixel enable optimizations of memory-access patterns.

Even though all functions in NPP will work with arbitrarily aligned images, best performance can only be achieved with well aligned image data. Any image data allocated with the NPP image allocators or the 2D memory allocators in the CUDA runtime, is well aligned.

Particularly on older CUDA capable GPUs it is likely that the performance decrease for misaligned data is substantial (orders of magnitude).

All image data passed to NPPI primitives requires a line step to be provided. It is important to keep in mind that this line step is always specified in terms of bytes, not pixels.

### 4.2.2 Parameter Names for Image Data

There are three general cases of image-data passing throughout NPP detailed in the following sections.

#### 4.2.2.1 Passing Source-Image Data

Those are images consumed by the algorithm.

##### 4.2.2.1.1 Source-Image Pointer

The source image data is generally passed via a pointer named

```
pSrc
```

The source image pointer is generally defined constant, enforcing that the primitive does not change any image data pointed to by that pointer. E.g.

```
npPiPrimitive_32s_C1R(const Npp32s * pSrc, ...)
```

In case the primitive consumes multiple images as inputs the source pointers are numbered like this:

```
pSrc1, pSrc2, ...
```

##### 4.2.2.1.2 Source-Planar-Image Pointer Array

The planar source image data is generally passed via an array of pointers named

```
pSrc[]
```

The planar source image pointer array is generally defined a constant array of constant pointers, enforcing that the primitive does not change any image data pointed to by those pointers. E.g.

```
nppiPrimitive_8u_P3R(const Npp8u * const pSrc[3], ...)
```

Each pointer in the array points to a different image plane.

#### **4.2.2.1.3 Source-Planar-Image Pointer**

The multiple plane source image data is passed via a set of pointers named

```
pSrc1, pSrc2, ...
```

The planar source image pointer is generally defined as one of a set of constant pointers with each pointer pointing to a different input image plane.

#### **4.2.2.1.4 Source-Image Line Step**

The source image line step is the number of bytes between successive rows in the image. The source image line step parameter is

```
nSrcStep
```

or in the case of multiple source images

```
nSrcStep1, nSrcStep2, ...
```

#### **4.2.2.1.5 Source-Planar-Image Line Step Array**

The source planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the input image. The source planar image line step array parameter is

```
rSrcStep[]
```

#### **4.2.2.1.6 Source-Planar-Image Line Step**

The source planar image line step is the number of bytes between successive rows in a particular plane of the multiplane input image. The source planar image line step parameter is

```
nSrcStep1, nSrcStep2, ...
```

#### **4.2.2.2 Passing Destination-Image Data**

Those are images produced by the algorithm.

#### 4.2.2.2.1 Destination-Image Pointer

The destination image data is generally passed via a pointer named

```
pDst
```

In case the primitive generates multiple images as outputs the destination pointers are numbered like this:

```
pDst1, pDst2, ...
```

#### 4.2.2.2.2 Destination-Planar-Image Pointer Array

The planar destination image data pointers are generally passed via an array of pointers named

```
pDst[]
```

Each pointer in the array points to a different image plane.

#### 4.2.2.2.3 Destination-Planar-Image Pointer

The destination planar image data is generally passed via a pointer to each plane of a multiplane output image named

```
pDst1, pDst2, ...
```

#### 4.2.2.2.4 Destination-Image Line Step

The destination image line step parameter is

```
nDstStep
```

or in the case of multiple destination images

```
nDstStep1, nDstStep2, ...
```

#### 4.2.2.2.5 Destination-Planar-Image Line Step Array

The destination planar image line step array is an array where each element of the array contains the number of bytes between successive rows for a particular plane in the output image. The destination planar image line step array parameter is

```
rDstStep[]
```

#### 4.2.2.2.6 Destination-Planar-Image Line Step

The destination planar image line step is the number of bytes between successive rows for a particular plane in a multiplane output image. The destination planar image line step parameter is

```
nDstStep1, nDstStep2, ...
```

### 4.2.2.3 Passing In-Place Image Data

#### 4.2.2.3.1 In-Place Image Pointer

In the case of in-place processing, source and destination are served by the same pointer and thus pointers to in-place image data are called:

`pSrcDst`

#### 4.2.2.3.2 In-Place-Image Line Step

The in-place line step parameter is

`nSrcDstStep`

### 4.2.2.4 Passing Mask-Image Data

Some image processing primitives have variants supporting [Masked Operation](#).

#### 4.2.2.4.1 Mask-Image Pointer

The mask-image data is generally passed via a pointer named

`pMask`

#### 4.2.2.4.2 Mask-Image Line Step

The mask-image line step parameter is

`nMaskStep`

### 4.2.2.5 Passing Channel-of-Interest Data

Some image processing primitives support [Channel-of-Interest API](#).

#### 4.2.2.5.1 Channel\_of\_Interest Number

The channel-of-interest data is generally an integer (either 1, 2, or 3):

`nCOI`

## 4.2.3 Image Data Alignment Requirements

NPP requires pixel data to adhere to certain alignment constraints: For 2 and 4 channel images the following alignment requirement holds: `data_pointer % (#channels * sizeof(channel type)) == 0`. E.g. a 4 channel image with underlying type [Npp8u](#) (8-bit unsigned) would require all pixels to fall on addresses that are multiples of 4 (4 channels \* 1 byte size).

As a logical consequence of all pixels being aligned to their natural size the image line steps of 2 and 4 channel images also need to be multiples of the pixel size.

1 and 3 channel images only require that pixel pointers are aligned to the underlying data type, i.e. `pData % sizeof(data type) == 0`. And consequentially line steps are also held to this requirement.

#### 4.2.4 Image Data Related Error Codes

All NPPI primitives operating on image data validate the image-data pointer for proper alignment and test that the point is not null. They also validate the line stride for proper alignment and guard against the step being less or equal to 0. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_STEP_ERROR` is returned if the data step is 0 or negative.
- `NPP_NOT_EVEN_STEP_ERROR` is returned if the line step is not a multiple of the pixel size for 2 and 4 channel images.
- `NPP_NULL_POINTER_ERROR` is returned if the image-data pointer is 0 (NULL).
- `NPP_ALIGNMENT_ERROR` if the image-data pointer address is not a multiple of the pixel size for 2 and 4 channel images.

### 4.3 Region-of-Interest (ROI)

In practice processing a rectangular sub-region of an image is often more common than processing complete images. The vast majority of NPP's image-processing primitives allow for processing of such sub regions also referred to as regions-of-interest or ROIs.

All primitives supporting ROI processing are marked by a "R" in their name suffix. In most cases the ROI is passed as a single `NppiSize` struct, which provides the width and height of the ROI. This raises the question how the primitive knows where in the image this rectangle of (width, height) is located. The "start pixel" of the ROI is implicitly given by the image-data pointer. I.e. instead of explicitly passing a pixel coordinate for the upper-right corner, the user simply offsets the image-data pointers to point to the first pixel of the ROI.

In practice this means that for an image (`pSrc`, `nSrcStep`) and the start-pixel of the ROI being at location (`xROI`, `yROI`), one would pass

`pSrcOffset = pSrc + yROI * nSrcStep + xROI * PixelSize;`

as the image-data source to the primitive. `PixelSize` is typically computed as

`PixelSize = NumberOfColorChannels * sizeof(PixelDataType).`

E.g. for a primitive like `nppiSet_16s_C4R()` we would have

- `NumberOfColorChannels == 4;`
- `sizeof(Npp16s) == 2;`
- and thus `PixelSize = 4 * 2 = 8;`

#### 4.3.1 ROI Related Error Codes

All NPPI primitives operating on ROIs of image data validate the ROI size and image's step size. Failed validation results in one of the following error codes being returned and the primitive not being executed:

- `NPP_SIZE_ERROR` is returned if either the ROI width or ROI height are negative.
- `NPP_STEP_ERROR` is returned if the ROI width exceeds the image's line step. In mathematical terms  $(\text{widthROI} * \text{PixelSize}) > \text{nLinStep}$  indicates an error.

## 4.4 Masked Operation

Some primitive support masked operation. An "M" in the suffix of those variants indicates masked operation. Primitives supporting masked operation consume an additional input image provided via a [Mask-Image Pointer](#) and [Mask-Image Line Step](#). The mask image is interpreted by these primitives as a boolean image. The values of type `Npp8u` are interpreted as boolean values where a values of 0 indicates false, any non-zero values true.

Unless otherwise indicated the operation is only performed on pixels where its spatially corresponding mask pixel is true (non-zero). E.g. a masked copy operation would only copy those pixels in the ROI that have corresponding non-zero mask pixels.

## 4.5 Channel-of-Interest API

Some primitives allow restricting operations to a single channel of interest within a multi-channel image. These primitives are suffixed with the letter "C" (after the channel information, e.g. `nppiCopy_8u_C3CR(...)`). The channel-of-interest is generally selected by offsetting the image-data pointer to point directly to the channel- of-interest rather than the base of the first pixel in the ROI. Some primitives also explicitly specify the selected channel number and pass it via an integer, e.g. `nppiMean_StdDev_8u_C3CR(...)`.

### 4.5.1 Select-Channel Source-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the source image. E.g. if `pSrc` is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel copy primitive one could copy the second channel of this source image into the first channel of a destination image given by `pDst` by offsetting the pointer by one:

```
nppiCopy_8u_C3CR(pSrc + 1, nSrcStep, pDst, nDstStep, oSizeROI);
```

### 4.5.2 Select-Channel Source-Image

Some primitives allow the user to select the channel-of-interest by specifying the channel number (`nCOI`). This approach is typically used in the image statistical functions. For example,

```
nppiMean_StdDev_8u_C3CR(pSrc, nSrcStep, oSizeROI, nCOI, pDeviceBuffer, pMean, pStdDev );
```

The channel-of-interest number can be either 1, 2, or 3.

### 4.5.3 Select-Channel Destination-Image Pointer

This is a pointer to the channel-of-interest within the first pixel of the destination image. E.g. if `pDst` is the pointer to the first pixel inside the ROI of a three channel image. Using the appropriate select-channel



copy primitive one could copy data into the second channel of this destination image from the first channel of a source image given by pSrc by offsetting the destination pointer by one:

```
nppiCopy_8u_C3CR(pSrc, nSrcStep, pDst + 1, nDstStep, oSizeROI);
```

## 4.6 Source-Image Sampling

A large number of NPP image-processing functions consume at least one source image and produce an output image (e.g. [nppiAddC\\_8u\\_C1RSfs\(\)](#) or [nppiFilterBox\\_8u\\_C1R\(\)](#)). All NPP functions falling into this category also operate on ROIs (see [Region-of-Interest \(ROI\)](#)) which for these functions should be considered to describe the destination ROI. In other words the ROI describes a rectangular region in the destination image and all pixels inside of this region are being written by the function in question.

In order to use such functions successfully it is important to understand how the user defined destination ROI affects which pixels in the input image(s) are being read by the algorithms. To simplify the discussion of ROI propagation (i.e. given a destination ROI, what are the ROIs in in the source(s)), it makes sense to distinguish two major cases:

1. Point-Wise Operations: These are primitives like [nppiAddC\\_8u\\_C1RSfs\(\)](#). Each output pixel requires exactly one input pixel to be read.
2. Neighborhood Operations: These are primitives like [nppiFilterBox\\_8u\\_C1R\(\)](#), which require a group of pixels from the source image(s) to be read in order to produce a single output.

### 4.6.1 Point-Wise Operations

As mentioned above, point-wise operations consume a single pixel from the input image (or a single pixel from each input image, if the operation in question has more than one input image) in order to produce a single output pixel.

### 4.6.2 Neighborhood Operations

In the case of neighborhood operations a number of input pixels (a "neighborhood" of pixels) is read in the input image (or images) in order to compute a single output pixel. All of the functions for [Filtering Functions](#) and [Morphological Operations](#) are neighborhood operations.

Most of these functions have parameters that affect the size and relative location of the neighborhood: a mask-size structure and an anchor-point structure. Both parameters are described in more detail in the next subsections.

#### 4.6.2.1 Mask-Size Parameter

Many NPP neighborhood operations allow the user to specify the size of the neighborhood via a parameter usually named oMaskSize of type [NppiSize](#). In those cases the neighborhood of pixels read from the source(s) is exactly the size of the mask. Assuming the mask is anchored at location (0, 0) (see [Anchor-Point Parameter](#) below) and has a size of (w, h), i.e.

```
assert(oMaskSize.w == w);
assert(oMaskSize.h == h);
assert(oAnchor.x == 0);
assert(oAnchor.y == 0);
```

a neighborhood operation would read the following source pixels in order to compute destination pixel  $D_{i,j}$ :

$$\begin{array}{cccc} S_{i,j} & S_{i,j+1} & \cdots & S_{i,j+w-1} \\ S_{i+1,j} & S_{i+1,j+1} & \cdots & S_{i+1,j+w-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i+h-1,j} & S_{i+h-1,j+1} & \cdots & S_{i+h-1,j+w-1} \end{array}$$

#### 4.6.2.2 Anchor-Point Parameter

Many NPP primitives performing neighborhood operations allow the user to specify the relative location of the neighborhood via a parameter usually named `oAnchor` of type [NppiPoint](#). Using the anchor a developer can choose the position of the mask (see [Mask-Size Parameter](#)) relative to current pixel index.

Using the same example as in [Mask-Size Parameter](#), but this time with an anchor position of (a, b):

```
assert(oMaskSize.w == w);
assert(oMaskSize.h == h);
assert(oAnchor.x == a);
assert(oAnchor.y == b);
```

the following pixels from the source image would be read:

$$\begin{array}{cccc} S_{i-a,j-b} & S_{i-a,j-b+1} & \cdots & S_{i-a,j-b+w-1} \\ S_{i-a+1,j-b} & S_{i-a+1,j-b+1} & \cdots & S_{i-a+1,j-b+w-1} \\ \vdots & \vdots & \ddots & \vdots \\ S_{i-a+h-1,j-b} & S_{i-a+h-1,j-b+1} & \cdots & S_{i-a+h-1,j-b+w-1} \end{array}$$

#### 4.6.2.3 Sampling Beyond Image Boundaries

NPP primitives in general and NPP neighborhood operations in particular require that all pixel locations read and written are valid and within the boundaries of the respective images. Sampling outside of the defined image data regions results in undefined behavior and may lead to system instability.

This poses a problem in practice: when processing full-size images one cannot choose the destination ROI to be the same size as the source image. Because neighborhood operations read pixels from an enlarged source ROI, the destination ROI must be shrunk so that the expanded source ROI does not exceed the source image's size.

For cases where this "shrinking" of the destination image size is unacceptable, NPP provides a set of border-expanding Copy primitives. E.g. [nppiCopyConstBorder\\_8u\\_C1R\(\)](#), [nppiCopyReplicateBorder\\_8u\\_C1R\(\)](#) and [nppiCopyWrapBorder\\_8u\\_C1R\(\)](#). The user can use these primitives to "expand" the source image's size using one of the three expansion modes. The expanded image can then be safely passed to a neighborhood operation producing a full-size result.

# Chapter 5

## Module Index

### 5.1 Modules

Here is a list of all modules:

|  |     |
|--|-----|
| NPP Core . . . . .                           | 31  |
| NPP Type Definitions and Constants . . . . . | 34  |
| Basic NPP Data Types . . . . .               | 47  |
| NPP Image Processing . . . . .               | 51  |
| Arithmetic and Logical Operations . . . . .  | 52  |
| Arithmetic Operations . . . . .              | 53  |
| AddC . . . . .                               | 55  |
| MulC . . . . .                               | 81  |
| MulCScale . . . . .                          | 107 |
| SubC . . . . .                               | 114 |
| DivC . . . . .                               | 140 |
| AbsDiffC . . . . .                           | 166 |
| Add . . . . .                                | 168 |
| AddSquare . . . . .                          | 197 |
| AddProduct . . . . .                         | 200 |
| AddWeighted . . . . .                        | 204 |
| Mul . . . . .                                | 208 |
| MulScale . . . . .                           | 237 |
| Sub . . . . .                                | 246 |
| Div . . . . .                                | 276 |
| Div_Round . . . . .                          | 305 |
| Abs . . . . .                                | 320 |
| AbsDiff . . . . .                            | 327 |
| Sqr . . . . .                                | 330 |
| Sqrt . . . . .                               | 344 |
| Ln . . . . .                                 | 356 |
| Exp . . . . .                                | 363 |
| Logical Operations . . . . .                 | 370 |
| AndC . . . . .                               | 371 |
| OrC . . . . .                                | 382 |
| XorC . . . . .                               | 393 |
| RShiftC . . . . .                            | 404 |
| LShiftC . . . . .                            | 421 |

|  |      |
|--|------|
| And . . . . .                              | 432  |
| Or . . . . .                               | 444  |
| Xor . . . . .                              | 456  |
| Not . . . . .                              | 468  |
| Alpha Composition . . . . .                | 472  |
| AlphaCompC . . . . .                       | 473  |
| AlphaPremulC . . . . .                     | 481  |
| AlphaComp . . . . .                        | 488  |
| AlphaPremul . . . . .                      | 495  |
| Color and Sampling Conversion . . . . .    | 497  |
| Color Model Conversion . . . . .           | 498  |
| Color Sampling Format Conversion . . . . . | 578  |
| Color Gamma Correction . . . . .           | 606  |
| Complement Color Key . . . . .             | 612  |
| Color Processing . . . . .                 | 615  |
| Compression . . . . .                      | 701  |
| Quantization Functions . . . . .           | 705  |
| Labeling and Segmentation . . . . .        | 711  |
| GraphCut . . . . .                         | 712  |
| Data Exchange and Initialization . . . . . | 719  |
| Set . . . . .                              | 720  |
| Copy . . . . .                             | 754  |
| Convert . . . . .                          | 801  |
| Scale . . . . .                            | 845  |
| Copy Constant Border . . . . .             | 860  |
| Copy Replicate Border . . . . .            | 873  |
| Copy Wrap Border . . . . .                 | 885  |
| Copy Sub-Pixel . . . . .                   | 898  |
| Duplicate Channel . . . . .                | 909  |
| Transpose . . . . .                        | 916  |
| Swap Channels . . . . .                    | 923  |
| Filtering Functions . . . . .              | 941  |
| 1D Linear Filter . . . . .                 | 989  |
| 1D Window Sum . . . . .                    | 1046 |
| Convolution . . . . .                      | 1057 |
| 2D Fixed Linear Filters . . . . .          | 1112 |
| Rank Filters . . . . .                     | 1121 |
| Fixed Filters . . . . .                    | 1151 |
| Geometry Transforms . . . . .              | 1183 |
| ResizeSqrPixel . . . . .                   | 1185 |
| Resize . . . . .                           | 1207 |
| Remap . . . . .                            | 1219 |
| Rotate . . . . .                           | 1241 |
| Mirror . . . . .                           | 1250 |
| Affine Transforms . . . . .                | 1267 |
| Perspective Transform . . . . .            | 1317 |
| Linear Transforms . . . . .                | 1363 |
| Fourier Transforms . . . . .               | 1364 |
| Morphological Operations . . . . .         | 1366 |
| Dilation . . . . .                         | 1367 |
| Erode . . . . .                            | 1374 |
| Dilate3x3 . . . . .                        | 1381 |
| Erode3x3 . . . . .                         | 1387 |

|   |      |
|---|------|
| Statistical Operations . . . . .            | 1393 |
| Sum . . . . .                               | 1460 |
| Min . . . . .                               | 1475 |
| MinIndx . . . . .                           | 1488 |
| Max . . . . .                               | 1502 |
| MaxIndx . . . . .                           | 1515 |
| MinMax . . . . .                            | 1529 |
| MinMaxIndx . . . . .                        | 1543 |
| Mean . . . . .                              | 1560 |
| Mean_StdDev . . . . .                       | 1581 |
| Image Norms . . . . .                       | 1597 |
| Norm_Inf . . . . .                          | 1599 |
| Norm_L1 . . . . .                           | 1621 |
| Norm_L2 . . . . .                           | 1642 |
| NormDiff_Inf . . . . .                      | 1663 |
| NormDiff_L1 . . . . .                       | 1686 |
| NormDiff_L2 . . . . .                       | 1709 |
| NormRel_Inf . . . . .                       | 1732 |
| NormRel_L1 . . . . .                        | 1755 |
| NormRel_L2 . . . . .                        | 1778 |
| DotProd . . . . .                           | 1801 |
| CountInRange. . . . .                       | 1826 |
| MaxEvery . . . . .                          | 1832 |
| MinEvery . . . . .                          | 1839 |
| Integral . . . . .                          | 1846 |
| SqrIntegral . . . . .                       | 1848 |
| RectStdDev . . . . .                        | 1851 |
| HistogramEven . . . . .                     | 1854 |
| HistogramRange . . . . .                    | 1867 |
| Image Proximity . . . . .                   | 1883 |
| SqrDistanceFull_Norm . . . . .              | 1886 |
| SqrDistanceSame_Norm . . . . .              | 1897 |
| SqrDistanceValid_Norm . . . . .             | 1908 |
| CrossCorrFull_Norm . . . . .                | 1919 |
| CrossCorrSame_Norm . . . . .                | 1930 |
| CrossCorrValid_Norm . . . . .               | 1941 |
| CrossCorrValid . . . . .                    | 1952 |
| CrossCorrFull_NormLevel . . . . .           | 1955 |
| CrossCorrSame_NormLevel . . . . .           | 1975 |
| CrossCorrValid_NormLevel . . . . .          | 1995 |
| Image Quality Index . . . . .               | 2015 |
| MaximumError . . . . .                      | 2024 |
| AverageError . . . . .                      | 2047 |
| MaximumRelativeError . . . . .              | 2070 |
| AverageRelativeError . . . . .              | 2094 |
| Memory Management . . . . .                 | 2118 |
| Threshold and Compare Operations . . . . .  | 2130 |
| Threshold Operations . . . . .              | 2131 |
| Compare Operations . . . . .                | 2220 |
| NPP Signal Processing . . . . .             | 2243 |
| Arithmetic and Logical Operations . . . . . | 2244 |
| Arithmetic Operations . . . . .             | 2245 |
| AddC . . . . .                              | 2247 |

|                                 |      |
|---------------------------------|------|
| AddProductC                     | 2256 |
| MulC                            | 2257 |
| SubC                            | 2267 |
| SubCRev                         | 2276 |
| DivC                            | 2285 |
| DivCRev                         | 2292 |
| Add                             | 2294 |
| AddProduct                      | 2306 |
| Mul                             | 2310 |
| Sub                             | 2323 |
| Div                             | 2333 |
| Div_Round                       | 2341 |
| Abs                             | 2344 |
| Sqr                             | 2347 |
| Sqrt                            | 2353 |
| Cubrt                           | 2361 |
| Exp                             | 2362 |
| Ln                              | 2366 |
| 10Log10                         | 2370 |
| SumLn                           | 2371 |
| Arctan                          | 2375 |
| Normalize                       | 2377 |
| Cauchy, CauchyD, and CauchyDD2  | 2380 |
| Logical And Shift Operations    | 2382 |
| AndC                            | 2383 |
| And                             | 2386 |
| OrC                             | 2389 |
| Or                              | 2392 |
| XorC                            | 2395 |
| Xor                             | 2398 |
| Not                             | 2401 |
| LShiftC                         | 2404 |
| RShiftC                         | 2408 |
| Conversion Functions            | 2412 |
| Convert                         | 2413 |
| Threshold                       | 2416 |
| Filtering Functions             | 2441 |
| Integral                        | 2442 |
| Initialization                  | 2443 |
| Set                             | 2444 |
| Zero                            | 2449 |
| Copy                            | 2453 |
| Statistical Functions           | 2457 |
| MinEvery And MaxEvery Functions | 2458 |
| Sum                             | 2462 |
| Maximum                         | 2469 |
| Minimum                         | 2479 |
| Mean                            | 2489 |
| Standard Deviation              | 2495 |
| Mean And Standard Deviation     | 2498 |
| Minimum_Maximum                 | 2502 |
| Infinity Norm                   | 2514 |
| L1 Norm                         | 2519 |

---

|                                |      |
|--------------------------------|------|
| L2 Norm . . . . .              | 2525 |
| Infinity Norm Diff . . . . .   | 2531 |
| L1 Norm Diff . . . . .         | 2536 |
| L2 Norm Diff . . . . .         | 2542 |
| Dot Product . . . . .          | 2548 |
| Count In Range . . . . .       | 2568 |
| Count Zero Crossings . . . . . | 2569 |
| MaximumError . . . . .         | 2571 |
| AverageError . . . . .         | 2582 |
| MaximumRelativeError . . . . . | 2593 |
| AverageRelativeError . . . . . | 2605 |
| Memory Management . . . . .    | 2617 |
| Malloc . . . . .               | 2618 |
| Free . . . . .                 | 2623 |





## Chapter 6

# Data Structure Index

### 6.1 Data Structures

Here are the data structures with brief descriptions:

|   |      |
|---|------|
| <a href="#">NPP_ALIGN_16</a> (Complex Number This struct represents a long long complex number ) . . . .                          | 2625 |
| <a href="#">NPP_ALIGN_8</a> (Complex Number This struct represents an unsigned int complex number ) . .                           | 2627 |
| <a href="#">NppiHaarBuffer</a> . . . . .  | 2629 |
| <a href="#">NppiHaarClassifier_32f</a> . . . . .  | 2630 |
| <a href="#">NppiPoint</a> (2D Point ) . . . . .   | 2631 |
| <a href="#">NppiRect</a> (2D Rectangle This struct contains position and size information of a rectangle in two space ) . . . . . | 2632 |
| <a href="#">NppiSize</a> (2D Size This struct typically represents the size of a a rectangular region in two space )              | 2633 |
| <a href="#">NppLibraryVersion</a> . . . . .   | 2634 |



# Chapter 7

## Module Documentation

### 7.1 NPP Core

Basic functions for library management, in particular library version and device property query functions.

#### Functions

- `const NppLibraryVersion * nppGetLibVersion` (void)  
*Get the NPP library version.*
- `NppGpuComputeCapability nppGetGpuComputeCapability` (void)  
*What CUDA compute model is supported by the active CUDA device?*
- `int nppGetGpuNumSMs` (void)  
*Get the number of Streaming Multiprocessors (SM) on the active CUDA device.*
- `int nppGetMaxThreadsPerBlock` (void)  
*Get the maximum number of threads per block on the active CUDA device.*
- `int nppGetMaxThreadsPerSM` (void)  
*Get the maximum number of threads per SM for the active GPU.*
- `const char * nppGetGpuName` (void)  
*Get the name of the active CUDA device.*
- `cudaStream_t nppGetStream` (void)  
*Get the NPP CUDA stream.*
- `void nppSetStream` (cudaStream\_t hStream)  
*Set the NPP CUDA stream.*

#### 7.1.1 Detailed Description

Basic functions for library management, in particular library version and device property query functions.

## 7.1.2 Function Documentation

### 7.1.2.1 `NppGpuComputeCapability nppGetGpuComputeCapability (void)`

What CUDA compute model is supported by the active CUDA device?

Before trying to call any NPP functions, the user should make a call this function to ensure that the current machine has a CUDA capable device.

**Returns:**

An enum value representing if a CUDA capable device was found and what level of compute capabilities it supports.

### 7.1.2.2 `const char* nppGetGpuName (void)`

Get the name of the active CUDA device.

**Returns:**

Name string of the active graphics-card/compute device in a system.

### 7.1.2.3 `int nppGetGpuNumSMs (void)`

Get the number of Streaming Multiprocessors (SM) on the active CUDA device.

**Returns:**

Number of SMs of the default CUDA device.

### 7.1.2.4 `const NppLibraryVersion* nppGetLibVersion (void)`

Get the NPP library version.

**Returns:**

A struct containing separate values for major and minor revision and build number.

### 7.1.2.5 `int nppGetMaxThreadsPerBlock (void)`

Get the maximum number of threads per block on the active CUDA device.

**Returns:**

Maximum number of threads per block on the active CUDA device.

**7.1.2.6 int nppGetMaxThreadsPerSM (void)**

Get the maximum number of threads per SM for the active GPU.

**Returns:**

Maximum number of threads per SM for the active GPU

**7.1.2.7 cudaStream\_t nppGetStream (void)**

Get the NPP CUDA stream.

NPP enables concurrent device tasks via a global stream state variable. The NPP stream by default is set to stream 0, i.e. non-concurrent mode. A user can set the NPP stream to any valid CUDA stream. All CUDA commands issued by NPP (e.g. kernels launched by the NPP library) are then issued to that NPP stream.

**7.1.2.8 void nppSetStream (cudaStream\_t *hStream*)**

Set the NPP CUDA stream.

**See also:**

[nppGetStream\(\)](#)

## 7.2 NPP Type Definitions and Constants

### Data Structures

- struct [NppLibraryVersion](#)
- struct [NppiPoint](#)  
*2D Point*
- struct [NppiSize](#)  
*2D Size This struct typically represents the size of a rectangular region in two space.*
- struct [NppiRect](#)  
*2D Rectangle This struct contains position and size information of a rectangle in two space.*
- struct [NppiHaarClassifier\\_32f](#)
- struct [NppiHaarBuffer](#)

### Modules

- [Basic NPP Data Types](#)

### Defines

- #define [NPP\\_MIN\\_8U](#) ( 0 )  
*Minimum 8-bit unsigned integer.*
- #define [NPP\\_MAX\\_8U](#) ( 255 )  
*Maximum 8-bit unsigned integer.*
- #define [NPP\\_MIN\\_16U](#) ( 0 )  
*Minimum 16-bit unsigned integer.*
- #define [NPP\\_MAX\\_16U](#) ( 65535 )  
*Maximum 16-bit unsigned integer.*
- #define [NPP\\_MIN\\_32U](#) ( 0 )  
*Minimum 32-bit unsigned integer.*
- #define [NPP\\_MAX\\_32U](#) ( 4294967295U )  
*Maximum 32-bit unsigned integer.*
- #define [NPP\\_MIN\\_64U](#) ( 0 )  
*Minimum 64-bit unsigned integer.*
- #define [NPP\\_MAX\\_64U](#) ( 18446744073709551615ULL )  
*Maximum 64-bit unsigned integer.*
- #define [NPP\\_MIN\\_8S](#) (-127 - 1 )  
*Minimum 8-bit signed integer.*

- #define `NPP_MAX_8S` ( 127 )  
*Maximum 8-bit signed integer.*
- #define `NPP_MIN_16S` (-32767 - 1 )  
*Minimum 16-bit signed integer.*
- #define `NPP_MAX_16S` ( 32767 )  
*Maximum 16-bit signed integer.*
- #define `NPP_MIN_32S` (-2147483647 - 1 )  
*Minimum 32-bit signed integer.*
- #define `NPP_MAX_32S` ( 2147483647 )  
*Maximum 32-bit signed integer.*
- #define `NPP_MAX_64S` ( 9223372036854775807LL )  
*Maximum 64-bit signed integer.*
- #define `NPP_MIN_64S` (-9223372036854775807LL - 1)  
*Minimum 64-bit signed integer.*
- #define `NPP_MINABS_32F` ( 1.175494351e-38f )  
*Smallest positive 32-bit floating point value.*
- #define `NPP_MAXABS_32F` ( 3.402823466e+38f )  
*Largest positive 32-bit floating point value.*
- #define `NPP_MINABS_64F` ( 2.2250738585072014e-308 )  
*Smallest positive 64-bit floating point value.*
- #define `NPP_MAXABS_64F` ( 1.7976931348623158e+308 )  
*Largest positive 64-bit floating point value.*

## Enumerations

- enum `NppiInterpolationMode` {  
`NPPI_INTER_UNDEFINED` = 0,  
`NPPI_INTER_NN` = 1,  
`NPPI_INTER_LINEAR` = 2,  
`NPPI_INTER_CUBIC` = 4,  
`NPPI_INTER_CUBIC2P_BSPLINE`,  
`NPPI_INTER_CUBIC2P_CATMULLROM`,  
`NPPI_INTER_CUBIC2P_B05C03`,  
`NPPI_INTER_SUPER` = 8,  
`NPPI_INTER_LANCZOS` = 16,  
`NPPI_SMOOTH_EDGE` = (1 << 31) }

*Filtering methods.*

- enum `NppiMaskSize` {  
`NPP_MASK_SIZE_1_X_3`,  
`NPP_MASK_SIZE_1_X_5`,  
`NPP_MASK_SIZE_3_X_1` = 100,  
`NPP_MASK_SIZE_5_X_1`,  
`NPP_MASK_SIZE_3_X_3` = 200,  
`NPP_MASK_SIZE_5_X_5` }

*Fixed filter-kernel sizes.*

- enum `NppStatus` {  
`NPP_NOT_SUPPORTED_MODE_ERROR` = -9999,  
`NPP_INVALID_HOST_POINTER_ERROR` = -1032,  
`NPP_INVALID_DEVICE_POINTER_ERROR` = -1031,  
`NPP_LUT_PALETTE_BITSIZE_ERROR` = -1030,  
`NPP_ZC_MODE_NOT_SUPPORTED_ERROR` = -1028,  
`NPP_NOT_SUFFICIENT_COMPUTE_CAPABILITY` = -1027,  
`NPP_TEXTURE_BIND_ERROR` = -1024,  
`NPP_WRONG_INTERSECTION_ROI_ERROR` = -1020,  
`NPP_HAAR_CLASSIFIER_PIXEL_MATCH_ERROR` = -1006,  
`NPP_MEMFREE_ERROR` = -1005,  
`NPP_MEMSET_ERROR` = -1004,  
`NPP_MEMCPY_ERROR` = -1003,  
`NPP_ALIGNMENT_ERROR` = -1002,  
`NPP_CUDA_KERNEL_EXECUTION_ERROR` = -1000,  
`NPP_ROUND_MODE_NOT_SUPPORTED_ERROR` = -213,  
`NPP_QUALITY_INDEX_ERROR` = -210,  
`NPP_RESIZE_NO_OPERATION_ERROR` = -201,  
`NPP_NOT_EVEN_STEP_ERROR` = -108,  
`NPP_HISTOGRAM_NUMBER_OF_LEVELS_ERROR` = -107,  
`NPP_LUT_NUMBER_OF_LEVELS_ERROR` = -106,  
`NPP_CHANNEL_ORDER_ERROR` = -60,  
`NPP_ZERO_MASK_VALUE_ERROR` = -59,  
`NPP_QUADRANGLE_ERROR` = -58,  
`NPP_RECTANGLE_ERROR` = -57,  
`NPP_COEFFICIENT_ERROR` = -56,  
`NPP_NUMBER_OF_CHANNELS_ERROR` = -53,  
`NPP_COI_ERROR` = -52,  
`NPP_DIVISOR_ERROR` = -51,  
`NPP_CHANNEL_ERROR` = -47,  
`NPP_STRIDE_ERROR` = -37,



```

NPP_ANCHOR_ERROR = -34,
NPP_MASK_SIZE_ERROR = -33,
NPP_RESIZE_FACTOR_ERROR = -23,
NPP_INTERPOLATION_ERROR = -22,
NPP_MIRROR_FLIP_ERROR = -21,
NPP_MOMENT_00_ZERO_ERROR = -20,
NPP_THRESHOLD_NEGATIVE_LEVEL_ERROR = -19,
NPP_THRESHOLD_ERROR = -18,
NPP_CONTEXT_MATCH_ERROR = -17,
NPP_FFT_FLAG_ERROR = -16,
NPP_FFT_ORDER_ERROR = -15,
NPP_STEP_ERROR = -14,
NPP_SCALE_RANGE_ERROR = -13,
NPP_DATA_TYPE_ERROR = -12,
NPP_OUT_OFF_RANGE_ERROR = -11,
NPP_DIVIDE_BY_ZERO_ERROR = -10,
NPP_MEMORY_ALLOCATION_ERR = -9,
NPP_NULL_POINTER_ERROR = -8,
NPP_RANGE_ERROR = -7,
NPP_SIZE_ERROR = -6,
NPP_BAD_ARGUMENT_ERROR = -5,
NPP_NO_MEMORY_ERROR = -4,
NPP_NOT_IMPLEMENTED_ERROR = -3,
NPP_ERROR = -2,
NPP_ERROR_RESERVED = -1,
NPP_NO_ERROR = 0,
NPP_SUCCESS = NPP_NO_ERROR,
NPP_NO_OPERATION_WARNING = 1,
NPP_DIVIDE_BY_ZERO_WARNING = 6,
NPP_AFFINE_QUAD_INCORRECT_WARNING = 28,
NPP_WRONG_INTERSECTION_ROI_WARNING = 29,
NPP_WRONG_INTERSECTION_QUAD_WARNING = 30,
NPP_DOUBLE_SIZE_WARNING = 35,
NPP_MISALIGNED_DST_ROI_WARNING = 10000 }

```

*Error Status Codes.*

- enum NppGpuComputeCapability {
 

```

NPP_CUDA_UNKNOWN_VERSION = -1,
NPP_CUDA_NOT_CAPABLE = 0,
NPP_CUDA_1_0 = 100,
NPP_CUDA_1_1 = 110,

```

```

NPP_CUDA_1_2 = 120,
NPP_CUDA_1_3 = 130,
NPP_CUDA_2_0 = 200,
NPP_CUDA_2_1 = 210,
NPP_CUDA_3_0 = 300,
NPP_CUDA_3_2 = 320,
NPP_CUDA_3_5 = 350,
NPP_CUDA_5_0 = 500 }
• enum NppiAxis {
    NPP_HORIZONTAL_AXIS,
    NPP_VERTICAL_AXIS,
    NPP_BOTH_AXIS }
• enum NppCmpOp {
    NPP_CMP_LESS,
    NPP_CMP_LESS_EQ,
    NPP_CMP_EQ,
    NPP_CMP_GREATER_EQ,
    NPP_CMP_GREATER }
• enum NppRoundMode {
    NPP_RND_NEAR,
    NPP_ROUND_NEAREST_TIES_TO_EVEN = NPP_RND_NEAR,
    NPP_RND_FINANCIAL,
    NPP_ROUND_NEAREST_TIES_AWAY_FROM_ZERO = NPP_RND_FINANCIAL,
    NPP_RND_ZERO,
    NPP_ROUND_TOWARD_ZERO = NPP_RND_ZERO }
    Rounding Modes.

• enum NppiBorderType {
    NPP_BORDER_UNDEFINED = 0,
    NPP_BORDER_NONE = NPP_BORDER_UNDEFINED,
    NPP_BORDER_CONSTANT = 1,
    NPP_BORDER_REPLICATE = 2,
    NPP_BORDER_WRAP = 3 }
• enum NppHintAlgorithm {
    NPP_ALG_HINT_NONE,
    NPP_ALG_HINT_FAST,
    NPP_ALG_HINT_ACCURATE }
• enum NppiAlphaOp {
    NPPI_OP_ALPHA_OVER,
    NPPI_OP_ALPHA_IN,
    NPPI_OP_ALPHA_OUT,
    NPPI_OP_ALPHA_ATOP,

```

```

    NPPI_OP_ALPHA_XOR,
    NPPI_OP_ALPHA_PLUS,
    NPPI_OP_ALPHA_OVER_PREMUL,
    NPPI_OP_ALPHA_IN_PREMUL,
    NPPI_OP_ALPHA_OUT_PREMUL,
    NPPI_OP_ALPHA_ATOP_PREMUL,
    NPPI_OP_ALPHA_XOR_PREMUL,
    NPPI_OP_ALPHA_PLUS_PREMUL,
    NPPI_OP_ALPHA_PREMUL }
• enum NppsZCType {
    nppZCR,
    nppZCXor,
    nppZCC }
• enum NppiHuffmanTableType {
    nppiDCTable,
    nppiACTable }

```

### 7.2.1 Define Documentation

#### 7.2.1.1 #define NPP\_MAX\_16S ( 32767 )

Maximum 16-bit signed integer.

#### 7.2.1.2 #define NPP\_MAX\_16U ( 65535 )

Maximum 16-bit unsigned integer.

#### 7.2.1.3 #define NPP\_MAX\_32S ( 2147483647 )

Maximum 32-bit signed integer.

#### 7.2.1.4 #define NPP\_MAX\_32U ( 4294967295U )

Maximum 32-bit unsigned integer.

#### 7.2.1.5 #define NPP\_MAX\_64S ( 9223372036854775807LL )

Maximum 64-bit signed integer.

#### 7.2.1.6 #define NPP\_MAX\_64U ( 18446744073709551615ULL )

Maximum 64-bit unsigned integer.

**7.2.1.7 #define NPP\_MAX\_8S ( 127 )**

Maximum 8-bit signed integer.

**7.2.1.8 #define NPP\_MAX\_8U ( 255 )**

Maximum 8-bit unsigned integer.

**7.2.1.9 #define NPP\_MAXABS\_32F ( 3.402823466e+38f )**

Largest positive 32-bit floating point value.

**7.2.1.10 #define NPP\_MAXABS\_64F ( 1.7976931348623158e+308 )**

Largest positive 64-bit floating point value.

**7.2.1.11 #define NPP\_MIN\_16S (-32767 - 1 )**

Minimum 16-bit signed integer.

**7.2.1.12 #define NPP\_MIN\_16U ( 0 )**

Minimum 16-bit unsigned integer.

**7.2.1.13 #define NPP\_MIN\_32S (-2147483647 - 1 )**

Minimum 32-bit signed integer.

**7.2.1.14 #define NPP\_MIN\_32U ( 0 )**

Minimum 32-bit unsigned integer.

**7.2.1.15 #define NPP\_MIN\_64S (-9223372036854775807LL - 1)**

Minimum 64-bit signed integer.

**7.2.1.16 #define NPP\_MIN\_64U ( 0 )**

Minimum 64-bit unsigned integer.

**7.2.1.17 #define NPP\_MIN\_8S (-127 - 1 )**

Minimum 8-bit signed integer.

**7.2.1.18 #define NPP\_MIN\_8U ( 0 )**

Minimum 8-bit unsigned integer.

**7.2.1.19 #define NPP\_MINABS\_32F ( 1.175494351e-38f )**

Smallest positive 32-bit floating point value.

**7.2.1.20 #define NPP\_MINABS\_64F ( 2.2250738585072014e-308 )**

Smallest positive 64-bit floating point value.

**7.2.2 Enumeration Type Documentation****7.2.2.1 enum NppCmpOp**

Enumerator:

*NPP\_CMP\_LESS*  
*NPP\_CMP\_LESS\_EQ*  
*NPP\_CMP\_EQ*  
*NPP\_CMP\_GREATER\_EQ*  
*NPP\_CMP\_GREATER*

**7.2.2.2 enum NppGpuComputeCapability**

Enumerator:

*NPP\_CUDA\_UNKNOWN\_VERSION* Indicates that the compute-capability query failed.  
*NPP\_CUDA\_NOT\_CAPABLE* Indicates that no CUDA capable device was found.  
*NPP\_CUDA\_1\_0* Indicates that CUDA 1.0 capable device is machine's default device.  
*NPP\_CUDA\_1\_1* Indicates that CUDA 1.1 capable device is machine's default device.  
*NPP\_CUDA\_1\_2* Indicates that CUDA 1.2 capable device is machine's default device.  
*NPP\_CUDA\_1\_3* Indicates that CUDA 1.3 capable device is machine's default device.  
*NPP\_CUDA\_2\_0* Indicates that CUDA 2.0 capable device is machine's default device.  
*NPP\_CUDA\_2\_1* Indicates that CUDA 2.1 capable device is machine's default device.  
*NPP\_CUDA\_3\_0* Indicates that CUDA 3.0 capable device is machine's default device.  
*NPP\_CUDA\_3\_2* Indicates that CUDA 3.2 capable device is machine's default device.  
*NPP\_CUDA\_3\_5* Indicates that CUDA 3.5 capable device is machine's default device.  
*NPP\_CUDA\_5\_0* Indicates that CUDA 5.0 or better is machine's default device.

**7.2.2.3 enum NppHintAlgorithm**

Enumerator:

*NPP\_ALG\_HINT\_NONE*  
*NPP\_ALG\_HINT\_FAST*  
*NPP\_ALG\_HINT\_ACCURATE*

#### 7.2.2.4 enum NppiAlphaOp

Enumerator:

*NPPI\_OP\_ALPHA\_OVER*  
*NPPI\_OP\_ALPHA\_IN*  
*NPPI\_OP\_ALPHA\_OUT*  
*NPPI\_OP\_ALPHA\_ATOP*  
*NPPI\_OP\_ALPHA\_XOR*  
*NPPI\_OP\_ALPHA\_PLUS*  
*NPPI\_OP\_ALPHA\_OVER\_PREMUL*  
*NPPI\_OP\_ALPHA\_IN\_PREMUL*  
*NPPI\_OP\_ALPHA\_OUT\_PREMUL*  
*NPPI\_OP\_ALPHA\_ATOP\_PREMUL*  
*NPPI\_OP\_ALPHA\_XOR\_PREMUL*  
*NPPI\_OP\_ALPHA\_PLUS\_PREMUL*  
*NPPI\_OP\_ALPHA\_PREMUL*

#### 7.2.2.5 enum NppiAxis

Enumerator:

*NPP\_HORIZONTAL\_AXIS*  
*NPP\_VERTICAL\_AXIS*  
*NPP\_BOTH\_AXIS*

#### 7.2.2.6 enum NppiBorderType

Enumerator:

*NPP\_BORDER\_UNDEFINED*  
*NPP\_BORDER\_NONE*  
*NPP\_BORDER\_CONSTANT*  
*NPP\_BORDER\_REPLICATE*  
*NPP\_BORDER\_WRAP*

#### 7.2.2.7 enum NppiHuffmanTableType

Enumerator:

*nppiDCTable* DC Table.  
*nppiACTable* AC Table.

**7.2.2.8 enum NppiInterpolationMode**

Filtering methods.

**Enumerator:**

***NPPI\_INTER\_UNDEFINED***  
***NPPI\_INTER\_NN*** Nearest neighbor filtering.  
***NPPI\_INTER\_LINEAR*** Linear interpolation.  
***NPPI\_INTER\_CUBIC*** Cubic interpolation.  
***NPPI\_INTER\_CUBIC2P\_BSPLINE*** Two-parameter cubic filter (B=1, C=0).  
***NPPI\_INTER\_CUBIC2P\_CATMULLROM*** Two-parameter cubic filter (B=0, C=1/2).  
***NPPI\_INTER\_CUBIC2P\_B05C03*** Two-parameter cubic filter (B=1/2, C=3/10).  
***NPPI\_INTER\_SUPER*** Super sampling.  
***NPPI\_INTER\_LANCZOS*** Lanczos filtering.  
***NPPI\_SMOOTH\_EDGE*** Smooth edge filtering.

**7.2.2.9 enum NppiMaskSize**

Fixed filter-kernel sizes.

**Enumerator:**

***NPP\_MASK\_SIZE\_1\_X\_3***  
***NPP\_MASK\_SIZE\_1\_X\_5***  
***NPP\_MASK\_SIZE\_3\_X\_1***  
***NPP\_MASK\_SIZE\_5\_X\_1***  
***NPP\_MASK\_SIZE\_3\_X\_3***  
***NPP\_MASK\_SIZE\_5\_X\_5***

**7.2.2.10 enum NppRoundMode**

Rounding Modes.

The enumerated rounding modes are used by a large number of NPP primitives to allow the user to specify the method by which fractional values are converted to integer values. Also see [Rounding Modes](#).

For NPP release 5.5 new names for the three rounding modes are introduced that are based on the naming conventions for rounding modes set forth in the IEEE-754 floating-point standard. Developers are encouraged to use the new, longer names to be future proof as the legacy names will be deprecated in subsequent NPP releases.

**Enumerator:**

***NPP\_RND\_NEAR*** Round to the nearest even integer.  
 All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e.  $\langle \text{integer} \rangle .5$ ) are rounded to the closest even integer. E.g.

- `roundNear(0.5) = 0`

- `roundNear(0.6) = 1`
- `roundNear(1.5) = 2`
- `roundNear(-1.5) = -2`

***NPP\_ROUND\_NEAREST\_TIES\_TO\_EVEN*** Alias name for [NPP\\_RND\\_NEAR](#).

***NPP\_RND\_FINANCIAL*** Round according to financial rule.

All fractional numbers are rounded to their nearest integer. The ambiguous cases (i.e. `<integer>.5`) are rounded away from zero. E.g.

- `roundFinancial(0.4) = 0`
- `roundFinancial(0.5) = 1`
- `roundFinancial(-1.5) = -2`

***NPP\_ROUND\_NEAREST\_TIES\_AWAY\_FROM\_ZERO*** Alias name for [NPP\\_RND\\_FINANCIAL](#).

***NPP\_RND\_ZERO*** Round towards zero (truncation).

All fractional numbers of the form `<integer>.<decimals>` are truncated to `<integer>`.

- `roundZero(1.5) = 1`
- `roundZero(1.9) = 1`
- `roundZero(-2.5) = -2`

***NPP\_ROUND\_TOWARD\_ZERO*** Alias name for [NPP\\_RND\\_ZERO](#).

#### 7.2.2.11 enum NppStatus

Error Status Codes.

Almost all NPP function return error-status information using these return codes. Negative return codes indicate errors, positive return codes indicate warnings, a return code of 0 indicates success.

**Enumerator:**

***NPP\_NOT\_SUPPORTED\_MODE\_ERROR***

***NPP\_INVALID\_HOST\_POINTER\_ERROR***

***NPP\_INVALID\_DEVICE\_POINTER\_ERROR***

***NPP\_LUT\_PALETTE\_BITSIZE\_ERROR***

***NPP\_ZC\_MODE\_NOT\_SUPPORTED\_ERROR*** ZeroCrossing mode not supported.

***NPP\_NOT\_SUFFICIENT\_COMPUTE\_CAPABILITY***

***NPP\_TEXTURE\_BIND\_ERROR***

***NPP\_WRONG\_INTERSECTION\_ROI\_ERROR***

***NPP\_HAAR\_CLASSIFIER\_PIXEL\_MATCH\_ERROR***

***NPP\_MEMFREE\_ERROR***

***NPP\_MEMSET\_ERROR***

***NPP\_MEMCPY\_ERROR***

***NPP\_ALIGNMENT\_ERROR***

***NPP\_CUDA\_KERNEL\_EXECUTION\_ERROR***

***NPP\_ROUND\_MODE\_NOT\_SUPPORTED\_ERROR*** Unsupported round mode.

***NPP\_QUALITY\_INDEX\_ERROR*** Image pixels are constant for quality index.



***NPP\_RESIZE\_NO\_OPERATION\_ERROR*** One of the output image dimensions is less than 1 pixel.

***NPP\_NOT\_EVEN\_STEP\_ERROR*** Step value is not pixel multiple.

***NPP\_HISTOGRAM\_NUMBER\_OF\_LEVELS\_ERROR*** Number of levels for histogram is less than 2.

***NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR*** Number of levels for LUT is less than 2.

***NPP\_CHANNEL\_ORDER\_ERROR*** Wrong order of the destination channels.

***NPP\_ZERO\_MASK\_VALUE\_ERROR*** All values of the mask are zero.

***NPP\_QUADRANGLE\_ERROR*** The quadrangle is nonconvex or degenerates into triangle, line or point.

***NPP\_RECTANGLE\_ERROR*** Size of the rectangle region is less than or equal to 1.

***NPP\_COEFFICIENT\_ERROR*** Unallowable values of the transformation coefficients.

***NPP\_NUMBER\_OF\_CHANNELS\_ERROR*** Bad or unsupported number of channels.

***NPP\_COI\_ERROR*** Channel of interest is not 1, 2, or 3.

***NPP\_DIVISOR\_ERROR*** Divisor is equal to zero.

***NPP\_CHANNEL\_ERROR*** Illegal channel index.

***NPP\_STRIDE\_ERROR*** Stride is less than the row length.

***NPP\_ANCHOR\_ERROR*** Anchor point is outside mask.

***NPP\_MASK\_SIZE\_ERROR*** Lower bound is larger than upper bound.

***NPP\_RESIZE\_FACTOR\_ERROR***

***NPP\_INTERPOLATION\_ERROR***

***NPP\_MIRROR\_FLIP\_ERROR***

***NPP\_MOMENT\_00\_ZERO\_ERROR***

***NPP\_THRESHOLD\_NEGATIVE\_LEVEL\_ERROR***

***NPP\_THRESHOLD\_ERROR***

***NPP\_CONTEXT\_MATCH\_ERROR***

***NPP\_FFT\_FLAG\_ERROR***

***NPP\_FFT\_ORDER\_ERROR***

***NPP\_STEP\_ERROR*** Step is less or equal zero.

***NPP\_SCALE\_RANGE\_ERROR***

***NPP\_DATA\_TYPE\_ERROR***

***NPP\_OUT\_OFF\_RANGE\_ERROR***

***NPP\_DIVIDE\_BY\_ZERO\_ERROR***

***NPP\_MEMORY\_ALLOCATION\_ERR***

***NPP\_NULL\_POINTER\_ERROR***

***NPP\_RANGE\_ERROR***

***NPP\_SIZE\_ERROR***

***NPP\_BAD\_ARGUMENT\_ERROR***

***NPP\_NO\_MEMORY\_ERROR***

***NPP\_NOT\_IMPLEMENTED\_ERROR***

***NPP\_ERROR***

***NPP\_ERROR\_RESERVED***

***NPP\_NO\_ERROR*** Error free operation.

***NPP\_SUCCESS*** Successful operation (same as ***NPP\_NO\_ERROR***).

***NPP\_NO\_OPERATION\_WARNING*** Indicates that no operation was performed.

***NPP\_DIVIDE\_BY\_ZERO\_WARNING*** Divisor is zero however does not terminate the execution.

***NPP\_AFFINE\_QUAD\_INCORRECT\_WARNING*** Indicates that the quadrangle passed to one of affine warping functions doesn't have necessary properties.

First 3 vertices are used, the fourth vertex discarded.

***NPP\_WRONG\_INTERSECTION\_ROI\_WARNING*** The given ROI has no interestion with either the source or destination ROI.

Thus no operation was performed.

***NPP\_WRONG\_INTERSECTION\_QUAD\_WARNING*** The given quadrangle has no intersection with either the source or destination ROI.

Thus no operation was performed.

***NPP\_DOUBLE\_SIZE\_WARNING*** Image size isn't multiple of two.

Indicates that in case of 422/411/420 sampling the ROI width/height was modified for proper processing.

***NPP\_MISALIGNED\_DST\_ROI\_WARNING*** Speed reduction due to uncoalesced memory accesses warning.

#### 7.2.2.12 enum NppsZCType

**Enumerator:**

***nppZCR*** sign change

***nppZCXor*** sign change XOR

***nppZCC*** sign change count\_0

## 7.3 Basic NPP Data Types

### Data Structures

- struct [NPP\\_ALIGN\\_8](#)  
*Complex Number This struct represents an unsigned int complex number.*
- struct [NPP\\_ALIGN\\_16](#)  
*Complex Number This struct represents a long long complex number.*

### Typedefs

- typedef unsigned char [Npp8u](#)  
*8-bit unsigned chars*
- typedef signed char [Npp8s](#)  
*8-bit signed chars*
- typedef unsigned short [Npp16u](#)  
*16-bit unsigned integers*
- typedef short [Npp16s](#)  
*16-bit signed integers*
- typedef unsigned int [Npp32u](#)  
*32-bit unsigned integers*
- typedef int [Npp32s](#)  
*32-bit signed integers*
- typedef unsigned long long [Npp64u](#)  
*64-bit unsigned integers*
- typedef long long [Npp64s](#)  
*64-bit signed integers*
- typedef float [Npp32f](#)  
*32-bit (IEEE) floating-point numbers*
- typedef double [Npp64f](#)  
*64-bit floating-point numbers*
- typedef struct [NPP\\_ALIGN\\_8](#) [Npp32uc](#)  
*Complex Number This struct represents an unsigned int complex number.*
- typedef struct [NPP\\_ALIGN\\_8](#) [Npp32sc](#)  
*Complex Number This struct represents a signed int complex number.*

- typedef struct [NPP\\_ALIGN\\_8 Npp32fc](#)  
*Complex Number This struct represents a single floating-point complex number.*
- typedef struct [NPP\\_ALIGN\\_16 Npp64sc](#)  
*Complex Number This struct represents a long long complex number.*
- typedef struct [NPP\\_ALIGN\\_16 Npp64fc](#)  
*Complex Number This struct represents a double floating-point complex number.*

## Functions

- struct [\\_\\_align\\_\\_](#) (2)  
*Complex Number This struct represents an unsigned char complex number.*
- struct [\\_\\_align\\_\\_](#) (4)  
*Complex Number This struct represents an unsigned short complex number.*

## Variables

- [Npp8uc](#)
- [Npp16uc](#)
- [Npp16sc](#)

### 7.3.1 Typedef Documentation

#### 7.3.1.1 typedef short Npp16s

16-bit signed integers

#### 7.3.1.2 typedef unsigned short Npp16u

16-bit unsigned integers

#### 7.3.1.3 typedef float Npp32f

32-bit (IEEE) floating-point numbers

#### 7.3.1.4 typedef struct NPP\_ALIGN\_8 Npp32fc

Complex Number This struct represents a single floating-point complex number.

#### 7.3.1.5 typedef int Npp32s

32-bit signed integers

**7.3.1.6 typedef struct NPP\_ALIGN\_8 Npp32sc**

Complex Number This struct represents a signed int complex number.

**7.3.1.7 typedef unsigned int Npp32u**

32-bit unsigned integers

**7.3.1.8 typedef struct NPP\_ALIGN\_8 Npp32uc**

Complex Number This struct represents an unsigned int complex number.

**7.3.1.9 typedef double Npp64f**

64-bit floating-point numbers

**7.3.1.10 typedef struct NPP\_ALIGN\_16 Npp64fc**

Complex Number This struct represents a double floating-point complex number.

**7.3.1.11 typedef long long Npp64s**

64-bit signed integers

**7.3.1.12 typedef struct NPP\_ALIGN\_16 Npp64sc**

Complex Number This struct represents a long long complex number.

**7.3.1.13 typedef unsigned long long Npp64u**

64-bit unsigned integers

**7.3.1.14 typedef signed char Npp8s**

8-bit signed chars

**7.3.1.15 typedef unsigned char Npp8u**

8-bit unsigned chars

**7.3.2 Function Documentation****7.3.2.1 struct \_\_align\_\_ (4) [read]**

Complex Number This struct represents an unsigned short complex number.

Complex Number This struct represents a short complex number.

< Real part

< Imaginary part

< Real part

< Imaginary part

#### **7.3.2.2 struct \_\_align\_\_ (2) [read]**

Complex Number This struct represents an unsigned char complex number.

< Real part

< Imaginary part

### **7.3.3 Variable Documentation**

#### **7.3.3.1 Npp16sc**

#### **7.3.3.2 Npp16uc**

#### **7.3.3.3 Npp8uc**

## 7.4 NPP Image Processing

### Modules

- [Arithmetic and Logical Operations](#)  
*Routines manipulating an image's color model and sampling format.*
- [Color and Sampling Conversion](#)  
*Routines manipulating an image's color model and sampling format.*
- [Compression](#)  
*Image compression primitives.*
- [Labeling and Segmentation](#)  
*Pixel labeling and image segmentation operations.*
- [Data Exchange and Initialization](#)  
*Primitives for initializing, copying and converting image data.*
- [Filtering Functions](#)  
*Linear and non-linear image filtering functions.*
- [Geometry Transforms](#)  
*Routines manipulating an image's geometry.*
- [Linear Transforms](#)  
*Linear image transformations.*
- [Morphological Operations](#)  
*Morphological image operations.*
- [Statistical Operations](#)  
*Primitives for computing the statistical properties of an image.*
- [Memory Management](#)  
*Routines for allocating and deallocating pitched image storage.*
- [Threshold and Compare Operations](#)  
*Methods for pixel-wise threshold and compare operations.*

## 7.5 Arithmetic and Logical Operations

### Modules

- [Arithmetic Operations](#)
- [Logical Operations](#)
- [Alpha Composition](#)



## 7.6 Arithmetic Operations

### Modules

- [AddC](#)  
*Adds a constant value to each pixel of an image.*
- [MulC](#)  
*Multiplies each pixel of an image by a constant value.*
- [MulCScale](#)  
*Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.*
- [SubC](#)  
*Subtracts a constant value from each pixel of an image.*
- [DivC](#)  
*Divides each pixel of an image by a constant value.*
- [AbsDiffC](#)  
*Determines absolute difference between each pixel of an image and a constant value.*
- [Add](#)  
*Pixel by pixel addition of two images.*
- [AddSquare](#)  
*Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.*
- [AddProduct](#)  
*Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.*
- [AddWeighted](#)  
*Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.*
- [Mul](#)  
*Pixel by pixel multiply of two images.*
- [MulScale](#)  
*Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.*
- [Sub](#)  
*Pixel by pixel subtraction of two images.*
- [Div](#)  
*Pixel by pixel division of two images.*

- [Div\\_Round](#)

*Pixel by pixel division of two images using result rounding modes.*

- [Abs](#)

*Absolute value of each pixel value in an image.*

- [AbsDiff](#)

*Pixel by pixel absolute difference between two images.*

- [Sqr](#)

*Square each pixel in an image.*

- [Sqrt](#)

*Pixel by pixel square root of each pixel in an image.*

- [Ln](#)

*Pixel by pixel natural logarithm of each pixel in an image.*

- [Exp](#)

*Exponential value of each pixel in an image.*

## 7.7 AddC

Adds a constant value to each pixel of an image.

### Functions

- **NppStatus** **nppiAddC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_8u\_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_8u\_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_8u\_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_16u\_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiAddC\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C3IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_AC4IRSfs` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16u_C4IRSfs` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C1IRSfs` (const `Npp16s` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C3IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_AC4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_AC4IRSfs` (const `Npp16s` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.*

- `NppStatus nppiAddC_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` nConstant, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_32sc_C1IRSfs` (const `Npp32sc` nConstant, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_32sc_C3IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_32sc_AC4IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.*
- `NppStatus nppiAddC_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` nConstant, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image add constant.*
- `NppStatus nppiAddC_32f_C1IR` (const `Npp32f` nConstant, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image add constant.*
- `NppStatus nppiAddC_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image add constant.*
- `NppStatus nppiAddC_32f_C3IR` (const `Npp32f` aConstants[3], `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel in place image add constant.*
- `NppStatus nppiAddC_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image add constant.*
- `NppStatus nppiAddC_32f_AC4IR` (const `Npp32f` aConstants[3], `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel with unmodified alpha in place image add constant.*

- `NppStatus nppiAddC_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` aConstants[4], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel image add constant.*

- `NppStatus nppiAddC_32f_C4IR` (const `Npp32f` aConstants[4], `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel in place image add constant.*

- `NppStatus nppiAddC_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` nConstant, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*

- `NppStatus nppiAddC_32fc_C1IR` (const `Npp32fc` nConstant, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*

- `NppStatus nppiAddC_32fc_C3R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` aConstants[3], `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*

- `NppStatus nppiAddC_32fc_C3IR` (const `Npp32fc` aConstants[3], `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*

- `NppStatus nppiAddC_32fc_AC4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` aConstants[3], `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.*

- `NppStatus nppiAddC_32fc_AC4IR` (const `Npp32fc` aConstants[3], `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.*

- `NppStatus nppiAddC_32fc_C4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` aConstants[4], `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.*

- `NppStatus nppiAddC_32fc_C4IR` (const `Npp32fc` aConstants[4], `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.*

### 7.7.1 Detailed Description

Adds a constant value to each pixel of an image.

### 7.7.2 Function Documentation

#### 7.7.2.1 `NppStatus nppiAddC_16s_AC4IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.2.2 `NppStatus nppiAddC_16s_AC4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.7.2.3 `NppStatus nppiAddC_16s_C1IRSfs (const Npp16s nConstant, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.



**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.4 NppStatus nppiAddC\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.5 NppStatus nppiAddC\_16s\_C3IRSfs (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.6 NppStatus nppiAddC\_16s\_C3RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.7 NppStatus nppiAddC\_16s\_C4IRSfs (const Npp16s *aConstants*[4], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.8 NppStatus nppiAddC\_16s\_C4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[4], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.9 NppStatus npAddC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.10 NppStatus npAddC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.11 **NppStatus nppiAddC\_16sc\_C1IRSfs** (const Npp16sc *nConstant*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.12 **NppStatus nppiAddC\_16sc\_C1RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc *nConstant*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.13 **NppStatus nppiAddC\_16sc\_C3IRSfs** (const Npp16sc *aConstants*[3], Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.14 NppStatus nppiAddC\_16sc\_C3RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc *aConstants*[3], Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.15 NppStatus nppiAddC\_16u\_AC4IRSfs** (const Npp16u *aConstants*[3], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.16 **NppStatus nppiAddC\_16u\_AC4RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.17 **NppStatus nppiAddC\_16u\_C1IRSfs** (const Npp16u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* [Constant](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.18 **NppStatus nppiAddC\_16u\_C1RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nConstant* [Constant](#).  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.19 NppStatus npAddC\_16u\_C3IRSfs (const Npp16u aConstants[3], Npp16u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.20 NppStatus npAddC\_16u\_C3RSfs (const Npp16u \*pSrcI, int nSrcIStep, const Npp16u aConstants[3], Npp16u \*pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.21 **NppStatus nppiAddC\_16u\_C4IRSfs** (const Npp16u *aConstants*[4], Npp16u \**pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.22 **NppStatus nppiAddC\_16u\_C4RSfs** (const Npp16u \**pSrcI*, int *nSrcIStep*, const Npp16u *aConstants*[4], Npp16u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).  
*nSrcIStep* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.23 **NppStatus nppiAddC\_32f\_AC4IR** (const Npp32f *aConstants*[3], Npp32f \**pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.7.2.24 NppStatus nppiAddC\_32f\_AC4R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.25 NppStatus nppiAddC\_32f\_C1IR (const Npp32f *nConstant*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image add constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.26 NppStatus nppiAddC\_32f\_C1R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.27 **NppStatus nppiAddC\_32f\_C3IR** (const Npp32f *aConstants*[3], Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.28 **NppStatus nppiAddC\_32f\_C3R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.29 **NppStatus nppiAddC\_32f\_C4IR** (const Npp32f *aConstants*[4], Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.30 **NppStatus nppiAddC\_32f\_C4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[4], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.31 **NppStatus nppiAddC\_32fc\_AC4IR** (const Npp32fc *aConstants*[3], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.32 **NppStatus nppiAddC\_32fc\_AC4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.33 **NppStatus nppiAddC\_32fc\_C1IR** (const Npp32fc *nConstant*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.34 **NppStatus nppiAddC\_32fc\_C1R** (const Npp32fc \* *pSrcI*, int *nSrcIStep*, const Npp32fc *nConstant*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.35 **NppStatus nppiAddC\_32fc\_C3IR** (const Npp32fc *aConstants*[3], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.36 **NppStatus nppiAddC\_32fc\_C3R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.37 **NppStatus nppiAddC\_32fc\_C4IR** (const Npp32fc *aConstants*[4], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image add constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.38 **NppStatus nppiAddC\_32fc\_C4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[4], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image add constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.39 NppStatus nppiAddC\_32s\_C1IRSfs (const Npp32s *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* [Constant](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.40 NppStatus nppiAddC\_32s\_C1IRSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*nConstant* [Constant](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.41 NppStatus nppiAddC\_32s\_C3IRSfs (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 32-bit signed integer channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* [fixed size array of constant values, one per channel](#).

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.42 NppStatus nppiAddC\_32s\_C3RSfs (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed integer channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.43 NppStatus nppiAddC\_32sc\_AC4IRSfs (const Npp32sc aConstants[3], Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.44 NppStatus nppiAddC\_32sc\_AC4RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.45 NppStatus nppiAddC\_32sc\_C1IRSfs (const Npp32sc *nConstant*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.7.2.46 NppStatus nppiAddC\_32sc\_C1RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *nConstant*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.



*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.47 **NppStatus nppiAddC\_32sc\_C3IRSfs** (const Npp32sc *aConstants*[3], Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.48 **NppStatus nppiAddC\_32sc\_C3RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.7.2.49 **NppStatus nppiAddC\_8u\_AC4IRSfs** (const Npp8u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel..

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.50 **NppStatus nppiAddC\_8u\_AC4RSfs** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image add constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel..

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.7.2.51 **NppStatus nppiAddC\_8u\_C1IRSfs** (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.52** `NppStatus nppiAddC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.53** `NppStatus nppiAddC_8u_C3RSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel..

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.7.2.54** `NppStatus nppiAddC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel..  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.55 NppStatus nppiAddC\_8u\_C4IRSfs (const Npp8u *aConstants*[4], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image add constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.7.2.56 NppStatus nppiAddC\_8u\_C4RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel..  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.8 MulC

Multiplies each pixel of an image by a constant value.

### Functions

- **NppStatus nppiMulC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_8u\_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_16u\_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*
- **NppStatus nppiMulC\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16u\_C3IRSfs** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16u\_AC4IRSfs** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16u\_C4IRSfs** (const **Npp16u** aConstants[4], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** nConstant, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16s\_C1IRSfs** (const **Npp16s** nConstant, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** aConstants[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16s\_C3IRSfs** (const **Npp16s** aConstants[3], **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** aConstants[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiMulC\_16s\_AC4IRSfs** (const **Npp16s** aConstants[3], **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.*
- `NppStatus nppiMulC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` nConstant, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32sc_C1IRSfs` (const `Npp32sc` nConstant, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32sc_C3IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32sc_AC4IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.*

- `NppStatus nppiMulC_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` nConstant, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point channel image multiply by constant.*

- `NppStatus nppiMulC_32f_C1IR` (const `Npp32f` nConstant, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point channel in place image multiply by constant.*

- `NppStatus nppiMulC_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit floating point channel image multiply by constant.*

- `NppStatus nppiMulC_32f_C3IR` (const `Npp32f` aConstants[3], `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 32-bit floating point channel in place image multiply by constant.*

- `NppStatus nppiMulC_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel with unmodified alpha image multiply by constant.*



- **NppStatus nppiMulC\_32f\_AC4IR** (const **Npp32f** aConstants[3], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.*
- **NppStatus nppiMulC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image multiply by constant.*
- **NppStatus nppiMulC\_32f\_C4IR** (const **Npp32f** aConstants[4], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C3IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_AC4IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[4], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.*
- **NppStatus nppiMulC\_32fc\_C4IR** (const **Npp32fc** aConstants[4], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.*

## 7.8.1 Detailed Description

Multiplies each pixel of an image by a constant value.

## 7.8.2 Function Documentation

### 7.8.2.1 `NppStatus nppiMulC_16s_AC4IRSfs (const Npp16s aConstants[3], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.2 `NppStatus nppiMulC_16s_AC4RSfs (const Npp16s * pSrcI, int nSrcIStep, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).

*nSrcIStep* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.3 NppStatus nppiMulC\_16s\_C1IRSfs (const Npp16s *nConstant*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.4 NppStatus nppiMulC\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.5 NppStatus nppiMulC\_16s\_C3IRSfs (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.6 `NppStatus nppiMulC_16s_C3RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[3], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.7 `NppStatus nppiMulC_16s_C4IRSfs (const Npp16s aConstants[4], Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.8 `NppStatus nppiMulC_16s_C4RSfs (const Npp16s * pSrc1, int nSrc1Step, const Npp16s aConstants[4], Npp16s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit signed short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.9 NppStatus nppiMulC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.10 NppStatus nppiMulC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.11 **NppStatus nppiMulC\_16sc\_C1IRSfs** (const Npp16sc *nConstant*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.12 **NppStatus nppiMulC\_16sc\_C1RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc *nConstant*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.13 **NppStatus nppiMulC\_16sc\_C3IRSfs** (const Npp16sc *aConstants*[3], Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.14 NppStatus nppiMulC\_16sc\_C3RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.15 NppStatus nppiMulC\_16u\_AC4IRSfs (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.16 **NppStatus nppiMulC\_16u\_AC4RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.17 **NppStatus nppiMulC\_16u\_C1IRSfs** (const Npp16u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.18 **NppStatus nppiMulC\_16u\_C1RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nConstant* Constant.



*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.19 NppStatus nppiMulC\_16u\_C3IRSfs (const Npp16u aConstants[3], Npp16u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.20 NppStatus nppiMulC\_16u\_C3RSfs (const Npp16u \*pSrcI, int nSrcIStep, const Npp16u aConstants[3], Npp16u \*pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.21 **NppStatus nppiMulC\_16u\_C4IRSfs** (const Npp16u *aConstants*[4], Npp16u \**pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.22 **NppStatus nppiMulC\_16u\_C4RSfs** (const Npp16u \**pSrcI*, int *nSrcIStep*, const Npp16u *aConstants*[4], Npp16u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel image multiply by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.23 **NppStatus nppiMulC\_32f\_AC4IR** (const Npp32f *aConstants*[3], Npp32f \**pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.24 **NppStatus nppiMulC\_32f\_AC4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image multiply by constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.25 **NppStatus nppiMulC\_32f\_C1IR** (const Npp32f *nConstant*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image multiply by constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.26 **NppStatus nppiMulC\_32f\_C1R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image multiply by constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.27 **NppStatus nppiMulC\_32f\_C3IR** (const Npp32f *aConstants*[3], Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.28 **NppStatus nppiMulC\_32f\_C3R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.29 **NppStatus nppiMulC\_32f\_C4IR** (const Npp32f *aConstants*[4], Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.30 **NppStatus nppiMulC\_32f\_C4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[4], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.31 **NppStatus nppiMulC\_32fc\_AC4IR** (const Npp32fc *aConstants*[3], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.32 **NppStatus nppiMulC\_32fc\_AC4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.33 `NppStatus nppiMulC_32fc_C1IR (const Npp32fc nConstant, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.34 `NppStatus nppiMulC_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc nConstant, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.35 `NppStatus nppiMulC_32fc_C3IR (const Npp32fc aConstants[3], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.8.2.36 **NppStatus nppiMulC\_32fc\_C3R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.37 **NppStatus nppiMulC\_32fc\_C4IR** (const Npp32fc *aConstants*[4], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image multiply by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.8.2.38 **NppStatus nppiMulC\_32fc\_C4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[4], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image multiply by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.39** `NppStatus nppiMulC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.40** `NppStatus nppiMulC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*nConstant* Constant.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.41** `NppStatus nppiMulC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image multiply by constant, scale, then clamp to saturated value.



**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.42 NppStatus nppiMulC\_32s\_C3RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.43 NppStatus nppiMulC\_32sc\_AC4IRSfs (const Npp32sc *aConstants*[3], Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.44 NppStatus nppiMulC\_32sc\_AC4RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.45 NppStatus nppiMulC\_32sc\_C1IRSfs (const Npp32sc *nConstant*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.8.2.46 NppStatus nppiMulC\_32sc\_C1RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *nConstant*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.47 NppStatus nppiMulC\_32sc\_C3IRSfs (const Npp32sc aConstants[3], Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.48 NppStatus nppiMulC\_32sc\_C3RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.49 **NppStatus nppiMulC\_8u\_AC4IRSfs** (const Npp8u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale, then clamp to saturated value.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.50 **NppStatus nppiMulC\_8u\_AC4RSfs** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.8.2.51 **NppStatus nppiMulC\_8u\_C1IRSfs** (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.52** `NppStatus nppiMulC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.53** `NppStatus nppiMulC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.8.2.54** `NppStatus nppiMulC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.55 **NppStatus nppiMulC\_8u\_C4IRSfs** (const Npp8u *aConstants*[4], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.8.2.56 **NppStatus nppiMulC\_8u\_C4RSfs** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image multiply by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.9 MulCScale

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

### Functions

- **NppStatus** **nppiMulCScale\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_8u\_C3IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_8u\_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_8u\_C4IR** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*

- **NppStatus** **nppiMulCScale\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_16u_C3IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_16u_AC4IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.*

- `NppStatus nppiMulCScale_16u_C4IR` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.*

## 7.9.1 Detailed Description

Multiplies each pixel of an image by a constant value then scales the result by the maximum value for the data bit width.

## 7.9.2 Function Documentation

### 7.9.2.1 `NppStatus nppiMulCScale_16u_AC4IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha in place image multiply by constant and scale by max bit width value.

#### Parameters:

**aConstants** fixed size array of constant values, one per channel.

**pSrcDst** In-Place Image Pointer.

**nSrcDstStep** In-Place-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes



### 7.9.2.2 NppStatus nppiMulCScale\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel with unmodified alpha image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.3 NppStatus nppiMulCScale\_16u\_C1IR (const Npp16u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.4 NppStatus nppiMulCScale\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.5 `NppStatus nppiMulCScale_16u_C3IR` (`const Npp16u aConstants[3]`, `Npp16u * pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

Three 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.6 `NppStatus nppiMulCScale_16u_C3R` (`const Npp16u * pSrc1`, `int nSrc1Step`, `const Npp16u aConstants[3]`, `Npp16u * pDst`, `int nDstStep`, `NppiSize oSizeROI`)

Three 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.7 `NppStatus nppiMulCScale_16u_C4IR` (`const Npp16u aConstants[4]`, `Npp16u * pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

Four 16-bit unsigned short channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.8 NppStatus nppiMulCScale\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[4], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.9 NppStatus nppiMulCScale\_8u\_AC4IR (const Npp8u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha in place image multiply by constant, scale and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.10 NppStatus nppiMulCScale\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.9.2.11 **NppStatus nppiMulCScale\_8u\_C1IR** (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.12 **NppStatus nppiMulCScale\_8u\_C1R** (const Npp8u \* *pSrcI*, int *nSrcIStep*, const Npp8u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.9.2.13 **NppStatus nppiMulCScale\_8u\_C3IR** (const Npp8u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel 8-bit unsigned char in place image multiply by constant and scale by max bit width value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.9.2.14 NppStatus nppiMulCScale\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.9.2.15 NppStatus nppiMulCScale\_8u\_C4IR (const Npp8u *aConstants*[4], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image multiply by constant and scale by max bit width value.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.9.2.16 NppStatus nppiMulCScale\_8u\_C4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image multiply by constant and scale by max bit width value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

## 7.10 SubC

Subtracts a constant value from each pixel of an image.

### Functions

- **NppStatus** **nppiSubC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_8u\_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_16u\_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiSubC\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16u\_C3IRSfs** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16u\_AC4IRSfs** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16u\_C4IRSfs** (const **Npp16u** aConstants[4], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** nConstant, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16s\_C1IRSfs** (const **Npp16s** nConstant, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** aConstants[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16s\_C3IRSfs** (const **Npp16s** aConstants[3], **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** aConstants[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16s\_AC4IRSfs** (const **Npp16s** aConstants[3], **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiSubC\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** aConstants[4], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.*

- `NppStatus nppiSubC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.*



- `NppStatus nppiSubC_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` nConstant, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_C1IRSfs` (const `Npp32sc` nConstant, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_C3IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` aConstants[3], `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32sc_AC4IRSfs` (const `Npp32sc` aConstants[3], `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.*
- `NppStatus nppiSubC_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` nConstant, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image subtract constant.*
- `NppStatus nppiSubC_32f_C1IR` (const `Npp32f` nConstant, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image subtract constant.*
- `NppStatus nppiSubC_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image subtract constant.*
- `NppStatus nppiSubC_32f_C3IR` (const `Npp32f` aConstants[3], `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel in place image subtract constant.*
- `NppStatus nppiSubC_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` aConstants[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image subtract constant.*
- `NppStatus nppiSubC_32f_AC4IR` (const `Npp32f` aConstants[3], `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel with unmodified alpha in place image subtract constant.*

- `NppStatus nppiSubC_32f_C4R` (const `Npp32f *pSrc1`, int `nSrc1Step`, const `Npp32f aConstants[4]`, `Npp32f *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 32-bit floating point channel image subtract constant.*

- `NppStatus nppiSubC_32f_C4IR` (const `Npp32f aConstants[4]`, `Npp32f *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 32-bit floating point channel in place image subtract constant.*

- `NppStatus nppiSubC_32fc_C1R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc nConstant`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*

- `NppStatus nppiSubC_32fc_C1IR` (const `Npp32fc nConstant`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*

- `NppStatus nppiSubC_32fc_C3R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[3]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*

- `NppStatus nppiSubC_32fc_C3IR` (const `Npp32fc aConstants[3]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*

- `NppStatus nppiSubC_32fc_AC4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[3]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.*

- `NppStatus nppiSubC_32fc_AC4IR` (const `Npp32fc aConstants[3]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.*

- `NppStatus nppiSubC_32fc_C4R` (const `Npp32fc *pSrc1`, int `nSrc1Step`, const `Npp32fc aConstants[4]`, `Npp32fc *pDst`, int `nDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.*

- `NppStatus nppiSubC_32fc_C4IR` (const `Npp32fc aConstants[4]`, `Npp32fc *pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.*

### 7.10.1 Detailed Description

Subtracts a constant value from each pixel of an image.

### 7.10.2 Function Documentation

#### 7.10.2.1 **NppStatus nppiSubC\_16s\_AC4IRSfs** (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.2 **NppStatus nppiSubC\_16s\_AC4RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.3 **NppStatus nppiSubC\_16s\_C1IRSfs** (const Npp16s *nConstant*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.4 NppStatus nppiSubC\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.5 NppStatus nppiSubC\_16s\_C3IRSfs (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.10.2.6 NppStatus nppiSubC\_16s\_C3RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.7 NppStatus nppiSubC\_16s\_C4IRSfs (const Npp16s *aConstants*[4], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.8 NppStatus nppiSubC\_16s\_C4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[4], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.9 NppStatus nppiSubC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.10 NppStatus nppiSubC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.11 NppStatus nppiSubC\_16sc\_C1IRSfs (const Npp16sc *nConstant*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.12 NppStatus nppiSubC\_16sc\_C1RSfs (const Npp16sc \* *pSrcI*, int *nSrcIStep*, const Npp16sc *nConstant*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.13 NppStatus nppiSubC\_16sc\_C3IRSfs (const Npp16sc *aConstants*[3], Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.14** `NppStatus nppiSubC_16sc_C3RSfs (const Npp16sc *pSrcI, int nSrcIStep, const Npp16sc aConstants[3], Npp16sc *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.15** `NppStatus nppiSubC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.10.2.16** `NppStatus nppiSubC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.17** `NppStatus nppiSubC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* [Constant](#).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.18** `NppStatus nppiSubC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nConstant* [Constant](#).  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.19 NppStatus nppiSubC\_16u\_C3IRSfs (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.20 NppStatus nppiSubC\_16u\_C3RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.10.2.21 `NppStatus nppiSubC_16u_C4IRSfs (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.22 `NppStatus nppiSubC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.23 `NppStatus nppiSubC_32f_AC4IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image subtract constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.24 **NppStatus nppiSubC\_32f\_AC4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image subtract constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.25 **NppStatus nppiSubC\_32f\_C1IR** (const Npp32f *nConstant*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image subtract constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.26 **NppStatus nppiSubC\_32f\_C1R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image subtract constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.10.2.27 `NppStatus nppiSubC_32f_C3IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel in place image subtract constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.28 `NppStatus nppiSubC_32f_C3R (const Npp32f * pSrcI, int nSrcIStep, const Npp32f aConstants[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image subtract constant.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).  
*nSrcIStep* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.29 `NppStatus nppiSubC_32f_C4IR (const Npp32f aConstants[4], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel in place image subtract constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.30 **NppStatus nppiSubC\_32f\_C4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[4], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image subtract constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.31 **NppStatus nppiSubC\_32fc\_AC4IR** (const Npp32fc *aConstants*[3], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image subtract constant.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.32 **NppStatus nppiSubC\_32fc\_AC4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image subtract constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.33 NppStatus nppiSubC\_32fc\_C1IR (const Npp32fc *nConstant*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.34 NppStatus nppiSubC\_32fc\_C1R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *nConstant*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.10.2.35 NppStatus nppiSubC\_32fc\_C3IR (const Npp32fc *aConstants*[3], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.36 **NppStatus nppiSubC\_32fc\_C3R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[3], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.37 **NppStatus nppiSubC\_32fc\_C4IR** (const Npp32fc *aConstants*[4], Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image subtract constant.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.38 **NppStatus nppiSubC\_32fc\_C4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc *aConstants*[4], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image subtract constant.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).



*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.39 NppStatus nppiSubC\_32s\_C1IRSfs (const Npp32s *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.40 NppStatus nppiSubC\_32s\_C1RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.41 NppStatus nppiSubC\_32s\_C3IRSfs (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 32-bit signed integer channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.42** `NppStatus nppiSubC_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[3], Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.43** `NppStatus nppiSubC_32sc_AC4IRSfs (const Npp32sc aConstants[3], Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.44** `NppStatus nppiSubC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.45** `NppStatus nppiSubC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.10.2.46** `NppStatus nppiSubC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.47 NppStatus npSubC\_32sc\_C3IRSfs (const Npp32sc aConstants[3], Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.48 NppStatus npSubC\_32sc\_C3RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.49 NppStatus nppiSubC\_8u\_AC4IRSfs (const Npp8u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image subtract constant, scale, then clamp to saturated value.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.50 NppStatus nppiSubC\_8u\_AC4RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image subtract constant, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.10.2.51 NppStatus nppiSubC\_8u\_C1IRSfs (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.52** `NppStatus nppiSubC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.53** `NppStatus nppiSubC_8u_C3IRSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.10.2.54** `NppStatus nppiSubC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.55 NppStatus nppiSubC\_8u\_C4IRSfs (const Npp8u *aConstants*[4], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.10.2.56 NppStatus nppiSubC\_8u\_C4RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.11 DivC

Divides each pixel of an image by a constant value.

### Functions

- **NppStatus** **nppiDivC\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_8u\_C1IRSfs** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_8u\_C3IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_8u\_AC4IRSfs** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_8u\_C4IRSfs** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_16u\_C1IRSfs** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*
- **NppStatus** **nppiDivC\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)



*Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16u\_C3IRSfs** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16u\_AC4IRSfs** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16u\_C4IRSfs** (const **Npp16u** aConstants[4], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** nConstant, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16s\_C1IRSfs** (const **Npp16s** nConstant, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** aConstants[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16s\_C3IRSfs** (const **Npp16s** aConstants[3], **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** aConstants[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_16s\_AC4IRSfs** (const **Npp16s** aConstants[3], **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*

- `NppStatus nppiDivC_16s_C4RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` aConstants[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16s_C4IRSfs` (const `Npp16s` aConstants[4], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C1RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` nConstant, `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C1IRSfs` (const `Npp16sc` nConstant, `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C3RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_C3IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_AC4RSfs` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` aConstants[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_16sc_AC4IRSfs` (const `Npp16sc` aConstants[3], `Npp16sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C1RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C1IRSfs` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.*
- `NppStatus nppiDivC_32s_C3IRSfs` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** nConstant, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_32sc\_C1IRSfs** (const **Npp32sc** nConstant, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** aConstants[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_32sc\_C3IRSfs** (const **Npp32sc** aConstants[3], **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** aConstants[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_32sc\_AC4IRSfs** (const **Npp32sc** aConstants[3], **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.*

- **NppStatus** **nppiDivC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** nConstant, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel image divided by constant.*

- **NppStatus** **nppiDivC\_32f\_C1IR** (const **Npp32f** nConstant, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 32-bit floating point channel in place image divided by constant.*

- **NppStatus** **nppiDivC\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 32-bit floating point channel image divided by constant.*

- **NppStatus** **nppiDivC\_32f\_C3IR** (const **Npp32f** aConstants[3], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Three 32-bit floating point channel in place image divided by constant.*

- **NppStatus** **nppiDivC\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point channel with unmodified alpha image divided by constant.*

- **NppStatus nppiDivC\_32f\_AC4IR** (const **Npp32f** aConstants[3], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image divided by constant.*
- **NppStatus nppiDivC\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** aConstants[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image divided by constant.*
- **NppStatus nppiDivC\_32f\_C4IR** (const **Npp32f** aConstants[4], **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** nConstant, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C1IR** (const **Npp32fc** nConstant, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C3IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.*
- **NppStatus nppiDivC\_32fc\_AC4IR** (const **Npp32fc** aConstants[3], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** aConstants[4], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.*
- **NppStatus nppiDivC\_32fc\_C4IR** (const **Npp32fc** aConstants[4], **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.*

### 7.11.1 Detailed Description

Divides each pixel of an image by a constant value.

### 7.11.2 Function Documentation

#### 7.11.2.1 **NppStatus nppiDivC\_16s\_AC4IRSfs** (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.11.2.2 **NppStatus nppiDivC\_16s\_AC4RSfs** (const Npp16s \* *pSrcI*, int *nSrcIStep*, const Npp16s *aConstants*[3], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcI* [Source-Image Pointer](#).

*nSrcIStep* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.3 **NppStatus nppiDivC\_16s\_C1RSfs** (const Npp16s *nConstant*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.4 **NppStatus nppiDivC\_16s\_C1RSfs** (const Npp16s \* *pSrcI*, int *nSrcIStep*, const Npp16s *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.5 **NppStatus nppiDivC\_16s\_C3IRSfs** (const Npp16s *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.6 NppStatus nppiDivC\_16s\_C3RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[3], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.7 NppStatus nppiDivC\_16s\_C4IRSfs (const Npp16s *aConstants*[4], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.8 NppStatus nppiDivC\_16s\_C4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s *aConstants*[4], Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.9 NppStatus nppiDivC\_16sc\_AC4IRSfs (const Npp16sc aConstants[3], Npp16sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.10 NppStatus nppiDivC\_16sc\_AC4RSfs (const Npp16sc \* pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*aConstants* fixed size array of constant values, one per channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.11.2.11 `NppStatus nppiDivC_16sc_C1IRSfs (const Npp16sc nConstant, Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.12 `NppStatus nppiDivC_16sc_C1RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc nConstant, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.13 `NppStatus nppiDivC_16sc_C3IRSfs (const Npp16sc aConstants[3], Npp16sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.14** `NppStatus nppiDivC_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc aConstants[3], Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.15** `NppStatus nppiDivC_16u_AC4IRSfs (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.11.2.16 `NppStatus nppiDivC_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.17 `NppStatus nppiDivC_16u_C1IRSfs (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.18 `NppStatus nppiDivC_16u_C1RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.19 NppStatus nppiDivC\_16u\_C3IRSfs (const Npp16u aConstants[3], Npp16u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.20 NppStatus nppiDivC\_16u\_C3RSfs (const Npp16u \*pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \*pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.11.2.21 `NppStatus nppiDivC_16u_C4IRSfs (const Npp16u aConstants[4], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.22 `NppStatus nppiDivC_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.23 `NppStatus nppiDivC_32f_AC4IR (const Npp32f aConstants[3], Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image divided by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.11.2.24 **NppStatus nppiDivC\_32f\_AC4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image divided by constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.25 **NppStatus nppiDivC\_32f\_C1IR** (const Npp32f *nConstant*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image divided by constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.26 **NppStatus nppiDivC\_32f\_C1R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *nConstant*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image divided by constant.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.11.2.27 **NppStatus nppiDivC\_32f\_C3IR** (const Npp32f *aConstants*[3], Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image divided by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.28 **NppStatus nppiDivC\_32f\_C3R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f *aConstants*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel image divided by constant.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.29 **NppStatus nppiDivC\_32f\_C4IR** (const Npp32f *aConstants*[4], Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image divided by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.30 `NppStatus nppiDivC_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f aConstants[4], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel image divided by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.31 `NppStatus nppiDivC_32fc_AC4IR (const Npp32fc aConstants[3], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha in place image divided by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.32 `NppStatus nppiDivC_32fc_AC4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel with unmodified alpha image divided by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.11.2.33 `NppStatus nppiDivC_32fc_C1IR` (const `Npp32fc nConstant`, `Npp32fc * pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.34 `NppStatus nppiDivC_32fc_C1R` (const `Npp32fc * pSrcI`, int `nSrcIStep`, const `Npp32fc nConstant`, `Npp32fc * pDst`, int `nDstStep`, `NppiSize oSizeROI`)

One 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.35 `NppStatus nppiDivC_32fc_C3IR` (const `Npp32fc aConstants[3]`, `Npp32fc * pSrcDst`, int `nSrcDstStep`, `NppiSize oSizeROI`)

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.36 `NppStatus nppiDivC_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.11.2.37 `NppStatus nppiDivC_32fc_C4IR (const Npp32fc aConstants[4], Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel in place image divided by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.11.2.38 `NppStatus nppiDivC_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc aConstants[4], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit complex floating point (32-bit floating point real, 32-bit floating point imaginary) channel image divided by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.39** `NppStatus nppiDivC_32s_C1IRSfs (const Npp32s nConstant, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* [Constant](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.40** `NppStatus nppiDivC_32s_C1RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s nConstant, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*nConstant* [Constant](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.41** `NppStatus nppiDivC_32s_C3IRSfs (const Npp32s aConstants[3], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.42 **NppStatus nppiDivC\_32s\_C3RSfs** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.43 **NppStatus nppiDivC\_32sc\_AC4IRSfs** (const Npp32sc *aConstants*[3], Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.44** `NppStatus nppiDivC_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc aConstants[3], Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.45** `NppStatus nppiDivC_32sc_C1IRSfs (const Npp32sc nConstant, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.11.2.46** `NppStatus nppiDivC_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc nConstant, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.47 **NppStatus nppiDivC\_32sc\_C3IRSfs** (const Npp32sc *aConstants*[3], Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.48 **NppStatus nppiDivC\_32sc\_C3RSfs** (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc *aConstants*[3], Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed complex integer (32-bit real, 32-bit imaginary) channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.11.2.49 NppStatus nppiDivC\_8u\_AC4IRSfs (const Npp8u aConstants[3], Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.50 NppStatus nppiDivC\_8u\_AC4RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.11.2.51 NppStatus nppiDivC\_8u\_C1IRSfs (const Npp8u nConstant, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.52** `NppStatus nppiDivC_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.53** `NppStatus nppiDivC_8u_C3RSfs (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel 8-bit unsigned char in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.11.2.54** `NppStatus nppiDivC_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.



**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.55 NppStatus nppiDivC\_8u\_C4IRSfs (const Npp8u *aConstants*[4], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.11.2.56 NppStatus nppiDivC\_8u\_C4RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.12 AbsDiffC

Determines absolute difference between each pixel of an image and a constant value.

### Functions

- **NppStatus nppiAbsDiffC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nConstant)  
*One 8-bit unsigned char channel image absolute difference with constant.*
- **NppStatus nppiAbsDiffC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp16u** nConstant)  
*One 16-bit unsigned short channel image absolute difference with constant.*
- **NppStatus nppiAbsDiffC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nConstant)  
*One 32-bit floating point channel image absolute difference with constant.*

### 7.12.1 Detailed Description

Determines absolute difference between each pixel of an image and a constant value.

### 7.12.2 Function Documentation

#### 7.12.2.1 **NppStatus nppiAbsDiffC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp16u** nConstant)

One 16-bit unsigned short channel image absolute difference with constant.

#### Parameters:

**pSrc1** Source-Image Pointer.  
**nSrc1Step** Source-Image Line Step.  
**nConstant** Constant.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.12.2.2 **NppStatus nppiAbsDiffC\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nConstant)

One 32-bit floating point channel image absolute difference with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.12.2.3 NppStatus nppiAbsDiffC\_8u\_C1R (const Npp8u \* pSrc1, int nSrc1Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, Npp8u nConstant)**

One 8-bit unsigned char channel image absolute difference with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.13 Add

Pixel by pixel addition of two images.

### Functions

- **NppStatus** **nppiAdd\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16u\_C1IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiAdd\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiAdd\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus** **nppiAdd\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*
- **NppStatus** **nppiAdd\_32s\_C1IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32s\_C3RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32s\_C3IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32sc\_C1IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32sc\_C3IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiAdd\_32sc\_AC4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiAdd_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point channel image addition.*

- `NppStatus nppiAdd_32f_C1IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point channel in place image addition.*

- `NppStatus nppiAdd_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit floating point channel image addition.*

- `NppStatus nppiAdd_32f_C3IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point channel in place image addition.*

- `NppStatus nppiAdd_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel with unmodified alpha image addition.*

- `NppStatus nppiAdd_32f_AC4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel with unmodified alpha in place image addition.*

- `NppStatus nppiAdd_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel image addition.*

- `NppStatus nppiAdd_32f_C4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel in place image addition.*

- `NppStatus nppiAdd_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*

- `NppStatus nppiAdd_32fc_C1IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*

- `NppStatus nppiAdd_32fc_C3R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*

- `NppStatus nppiAdd_32fc_C3IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)

*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*



- **NppStatus nppiAdd\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.*
- **NppStatus nppiAdd\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.*
- **NppStatus nppiAdd\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.*
- **NppStatus nppiAdd\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.*

### 7.13.1 Detailed Description

Pixel by pixel addition of two images.

### 7.13.2 Function Documentation

#### 7.13.2.1 **NppStatus nppiAdd\_16s\_AC4IRSfs** (const **Npp16s** \* pSrc, int nSrcStep, **Npp16s** \* pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pSrcDst** In-Place Image Pointer.  
**nSrcDstStep** In-Place-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).  
**nScaleFactor** Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.13.2.2 **NppStatus nppiAdd\_16s\_AC4RSfs** (const **Npp16s** \* pSrc1, int nSrc1Step, const **Npp16s** \* pSrc2, int nSrc2Step, **Npp16s** \* pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.3 NppStatus nppiAdd\_16s\_C1IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.4 NppStatus nppiAdd\_16s\_C1RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.5 NppStatus npAdd\_16s\_C3IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.6 NppStatus npAdd\_16s\_C3RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.7 **NppStatus nppiAdd\_16s\_C4IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.8 **NppStatus nppiAdd\_16s\_C4RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.9 **NppStatus nppiAdd\_16sc\_AC4IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.10 **NppStatus nppiAdd\_16sc\_AC4RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.11 **NppStatus nppiAdd\_16sc\_C1IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.12 NppStatus nppiAdd\_16sc\_C1RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.13 NppStatus nppiAdd\_16sc\_C3IRSfs (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.14 NppStatus nppiAdd\_16sc\_C3RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.15 **NppStatus nppiAdd\_16u\_AC4IRSfs** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.16 **NppStatus nppiAdd\_16u\_AC4RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.17 NppStatus npAdd\_16u\_C1RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.18 NppStatus npAdd\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.13.2.19 NppStatus npAdd\_16u\_C3IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.20 NppStatus npAdd\_16u\_C3RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Three 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.21 NppStatus npAdd\_16u\_C4IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.22** `NppStatus npplAdd_16u_C4RSfs (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.23** `NppStatus npplAdd_32f_AC4IR (const Npp32f *pSrc, int nSrcStep, Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.24 NppStatus nppiAdd\_32f\_AC4R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image addition.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.13.2.25 NppStatus nppiAdd\_32f\_C1IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image addition.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.13.2.26 NppStatus nppiAdd\_32f\_C1R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image addition.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.27 NppStatus npAdd\_32f\_C3IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.28 NppStatus npAdd\_32f\_C3R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.29 **NppStatus nppiAdd\_32f\_C4IR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image addition.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.30 **NppStatus nppiAdd\_32f\_C4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image addition.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.31 **NppStatus nppiAdd\_32fc\_AC4IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.32 **NppStatus nppiAdd\_32fc\_AC4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.13.2.33 **NppStatus nppiAdd\_32fc\_C1IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.13.2.34 **NppStatus nppiAdd\_32fc\_C1R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.35 NppStatus npAdd\_32fc\_C3IR (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.36 NppStatus npAdd\_32fc\_C3R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc \* pSrc2, int nSrc2Step, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.37** `NppStatus nppiAdd_32fc_C4IR (const Npp32fc * pSrc, int nSrcStep, Npp32fc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image addition.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.38** `NppStatus nppiAdd_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image addition.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.39** `NppStatus nppiAdd_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.



*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.40 NppStatus npAdd\_32s\_C1R (const Npp32s \*pSrc1, int nSrc1Step, const Npp32s \*pSrc2, int nSrc2Step, Npp32s \*pDst, int nDstStep, NppiSize oSizeROI)**

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image add. Add the pixel values of corresponding pixels in the ROI and write them to the output image.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.41 NppStatus npAdd\_32s\_C1RSfs (const Npp32s \*pSrc1, int nSrc1Step, const Npp32s \*pSrc2, int nSrc2Step, Npp32s \*pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.42 **NppStatus nppiAdd\_32s\_C3IRSfs** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.43 **NppStatus nppiAdd\_32s\_C3RSfs** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.44 **NppStatus nppiAdd\_32sc\_AC4IRSfs** (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.45 NppStatus npAdd\_32sc\_AC4RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.46 NppStatus npAdd\_32sc\_C1IRSfs (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.47 NppStatus npAdd\_32sc\_C1RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.48 NppStatus npAdd\_32sc\_C3RSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.49 NppStatus npAdd\_32sc\_C3RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.50 **NppStatus nppiAdd\_8u\_AC4IRSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.13.2.51 **NppStatus nppiAdd\_8u\_AC4RSfs** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.52** `NppStatus nppiAdd_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.13.2.53** `NppStatus nppiAdd_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.13.2.54 NppStatus nppiAdd\_8u\_C3IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.55 NppStatus nppiAdd\_8u\_C3RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.13.2.56 NppStatus nppiAdd\_8u\_C4IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.13.2.57** `NppStatus nppiAdd_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image addition, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



## 7.14 AddSquare

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

### Functions

- **NppStatus nppiAddSquare\_8u32f\_C1IMR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddSquare\_8u32f\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image squared then added to in place floating point destination image.*
- **NppStatus nppiAddSquare\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddSquare\_16u32f\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image squared then added to in place floating point destination image.*
- **NppStatus nppiAddSquare\_32f\_C1IMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddSquare\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image squared then added to in place floating point destination image.*

### 7.14.1 Detailed Description

Pixel by pixel addition of squared pixels from source image to floating point pixel values of destination image.

### 7.14.2 Function Documentation

#### 7.14.2.1 **NppStatus nppiAddSquare\_16u32f\_C1IMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

#### Parameters:

**pSrc** Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.14.2.2 NppStatus nppiAddSquare\_16u32f\_C1IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.14.2.3 NppStatus nppiAddSquare\_32f\_C1IMR (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.14.2.4 NppStatus nppiAddSquare\_32f\_C1IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.14.2.5 NppStatus nppiAddSquare\_8u32f\_C1IMR (const Npp8u \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel image squared then added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.14.2.6 NppStatus nppiAddSquare\_8u32f\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel image squared then added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.15 AddProduct

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

### Functions

- **NppStatus nppiAddProduct\_8u32f\_C1IMR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddProduct\_8u32f\_C1IR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image product added to in place floating point destination image.*
- **NppStatus nppiAddProduct\_16u32f\_C1IMR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddProduct\_16u32f\_C1IR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image product added to in place floating point destination image.*
- **NppStatus nppiAddProduct\_32f\_C1IMR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus nppiAddProduct\_32f\_C1IR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image product added to in place floating point destination image.*

### 7.15.1 Detailed Description

Pixel by pixel addition of product of pixels from two source images to floating point pixel values of destination image.

### 7.15.2 Function Documentation

- 7.15.2.1 NppStatus nppiAddProduct\_16u32f\_C1IMR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.2 NppStatus nppiAddProduct\_16u32f\_C1IR (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.3 NppStatus nppiAddProduct\_32f\_C1IMR (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, const Npp8u \* *pMask*, int *nMaskStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.4 NppStatus nppiAddProduct\_32f\_C1IR (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.15.2.5 NppStatus nppiAddProduct\_8u32f\_C1IMR (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, const Npp8u \* *pMask*, int *nMaskStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image product added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.15.2.6 NppStatus nppiAddProduct\_8u32f\_C1IR (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel image product added to in place floating point destination image.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.16 AddWeighted

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

### Functions

- **NppStatus** `nppiAddWeighted_8u32f_C1IMR` (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus** `nppiAddWeighted_8u32f_C1IR` (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.*
- **NppStatus** `nppiAddWeighted_16u32f_C1IMR` (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus** `nppiAddWeighted_16u32f_C1IR` (const **Npp16u** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.*
- **NppStatus** `nppiAddWeighted_32f_C1IMR` (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).*
- **NppStatus** `nppiAddWeighted_32f_C1IR` (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)  
*One 32-bit floating point channel alpha weighted image added to in place floating point destination image.*

### 7.16.1 Detailed Description

Pixel by pixel addition of alpha weighted pixel values from a source image to floating point pixel values of destination image.

### 7.16.2 Function Documentation

- 7.16.2.1** **NppStatus** `nppiAddWeighted_16u32f_C1IMR` (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **Npp32f** nAlpha)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).



**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.16.2.2 NppStatus nppiAddWeighted\_16u32f\_C11R (const Npp16u \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 16-bit unsigned short channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.16.2.3 NppStatus nppiAddWeighted\_32f\_C11MR (const Npp32f \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)

One 32-bit floating point channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.16.2.4 NppStatus npplAddWeighted\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)**

One 32-bit floating point channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.16.2.5 NppStatus npplAddWeighted\_8u32f\_C1IMR (const Npp8u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, Npp32f nAlpha)**

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image using filter mask (updates destination when mask is non-zero).

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.16.2.6 NppStatus nppiAddWeighted\_8u32f\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, Npp32f *nAlpha*)**

One 8-bit unsigned char channel alpha weighted image added to in place floating point destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAlpha* Alpha weight to be applied to source image pixels (0.0F to 1.0F)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.17 Mul

Pixel by pixel multiply of two images.

### Functions

- **NppStatus nppiMul\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiMul\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_16u_C1IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C3RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C3IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_AC4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_AC4IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C4RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16u_C4IRSfs` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C1RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C1IRSfs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C3RSfs` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_16s_C3IRSfs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*

- **NppStatus nppiMul\_32s\_C1IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32s\_C3RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32s\_C3IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32sc\_C1IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32sc\_C3IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiMul\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiMul_32sc_AC4IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiMul_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image multiplication.*
- `NppStatus nppiMul_32f_C1IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image multiplication.*
- `NppStatus nppiMul_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image multiplication.*
- `NppStatus nppiMul_32f_C3IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image multiplication.*
- `NppStatus nppiMul_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image multiplication.*
- `NppStatus nppiMul_32f_AC4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image multiplication.*
- `NppStatus nppiMul_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel image multiplication.*
- `NppStatus nppiMul_32f_C4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel in place image multiplication.*
- `NppStatus nppiMul_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*
- `NppStatus nppiMul_32fc_C1IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*
- `NppStatus nppiMul_32fc_C3R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*



- **NppStatus nppiMul\_32fc\_C3IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*
- **NppStatus nppiMul\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.*
- **NppStatus nppiMul\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.*
- **NppStatus nppiMul\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.*
- **NppStatus nppiMul\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.*

### 7.17.1 Detailed Description

Pixel by pixel multiply of two images.

### 7.17.2 Function Documentation

#### 7.17.2.1 **NppStatus nppiMul\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pSrcDst** In-Place Image Pointer.  
**nSrcDstStep** In-Place-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).  
**nScaleFactor** Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.2 NppStatus nppiMul\_16s\_AC4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.17.2.3 NppStatus nppiMul\_16s\_C1IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.17.2.4 NppStatus nppiMul\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.5 NppStatus nppiMul\_16s\_C3IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.6 NppStatus nppiMul\_16s\_C3RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.7 **NppStatus nppiMul\_16s\_C4IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.8 **NppStatus nppiMul\_16s\_C4RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.9 **NppStatus nppiMul\_16sc\_AC4IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.10 NppStatus nppiMul\_16sc\_AC4RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.11 NppStatus nppiMul\_16sc\_C1IRSfs (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.12 NppStatus nppiMul\_16sc\_C1RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.13 NppStatus nppiMul\_16sc\_C3IRSfs (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.14 NppStatus nppiMul\_16sc\_C3RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.15 NppStatus nppiMul\_16u\_AC4IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.16 NppStatus nppiMul\_16u\_AC4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit unsigned short channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.17 NppStatus nppiMul\_16u\_C1RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.18 NppStatus nppiMul\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.17.2.19 NppStatus nppiMul\_16u\_C3IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.20 NppStatus nppiMul\_16u\_C3RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.21 NppStatus nppiMul\_16u\_C4IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.22** `NppStatus nppiMul_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.23** `NppStatus nppiMul_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.24 **NppStatus nppiMul\_32f\_AC4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image multiplication.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.25 **NppStatus nppiMul\_32f\_C1IR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image multiplication.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.26 **NppStatus nppiMul\_32f\_C1R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image multiplication.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.27** `NppStatus nppiMul_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.28** `NppStatus nppiMul_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.29 **NppStatus nppiMul\_32f\_C4IR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel in place image multiplication.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.30 **NppStatus nppiMul\_32f\_C4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image multiplication.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.31 **NppStatus nppiMul\_32fc\_AC4IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.32 **NppStatus nppiMul\_32fc\_AC4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.17.2.33 **NppStatus nppiMul\_32fc\_C1IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.17.2.34 **NppStatus nppiMul\_32fc\_C1R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.35 NppStatus nppiMul\_32fc\_C3IR (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.36 NppStatus nppiMul\_32fc\_C3R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.37 **NppStatus nppiMul\_32fc\_C4IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image multiplication.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.38 **NppStatus nppiMul\_32fc\_C4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image multiplication.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.39 **NppStatus nppiMul\_32s\_C1IRSfs** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.



*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.40 NppStatus nppiMul\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

1 channel 32-bit image multiplication. Multiply corresponding pixels in ROI.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.41 NppStatus nppiMul\_32s\_C1RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.42 **NppStatus nppiMul\_32s\_C3IRSfs** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.43 **NppStatus nppiMul\_32s\_C3RSfs** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 32-bit signed integer channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.44 **NppStatus nppiMul\_32sc\_AC4IRSfs** (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.45** `NppStatus nppiMul_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.46** `NppStatus nppiMul_32sc_C1IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.47** `NppStatus nppiMul_32sc_C1RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.48** `NppStatus nppiMul_32sc_C3IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.49** `NppStatus nppiMul_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.50 NppStatus nppiMul\_8u\_AC4IRSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.17.2.51 NppStatus nppiMul\_8u\_AC4RSfs (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 8-bit unsigned char channel with unmodified alpha image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.52** `NppStatus nppiMul_8u_C1IRSfs (const Npp8u *pSrc, int nSrcStep, Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.53** `NppStatus nppiMul_8u_C1RSfs (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.54 NppStatus nppiMul\_8u\_C3IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.55 NppStatus nppiMul\_8u\_C3RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.17.2.56 NppStatus nppiMul\_8u\_C4IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.17.2.57** `NppStatus nppiMul_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image multiplication, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



## 7.18 MulScale

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

### Functions

- **NppStatus nppiMulScale\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C3IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_AC4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_8u\_C4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*
- **NppStatus nppiMulScale\_16u\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C3IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.*

- **NppStatus nppiMulScale\_16u\_C4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.*

### 7.18.1 Detailed Description

Pixel by pixel multiplies each pixel of two images then scales the result by the maximum value for the data bit width.

### 7.18.2 Function Documentation

#### 7.18.2.1 **NppStatus nppiMulScale\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

#### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pSrcDst** In-Place Image Pointer.

**nSrcDstStep** In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.2 NppStatus nppiMulScale\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 16-bit unsigned short channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.3 NppStatus nppiMulScale\_16u\_C11R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.18.2.4 **NppStatus nppiMulScale\_16u\_C1R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.18.2.5 **NppStatus nppiMulScale\_16u\_C3IR** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.18.2.6 **NppStatus nppiMulScale\_16u\_C3R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.7 NppStatus nppiMulScale\_16u\_C4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.8 NppStatus nppiMulScale\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.18.2.9 **NppStatus nppiMulScale\_8u\_AC4IR** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha in place image multiplication then scale by maximum value for pixel bit width.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.18.2.10 **NppStatus nppiMulScale\_8u\_AC4R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel with unmodified alpha image multiplication then scale by maximum value for pixel bit width.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.18.2.11 **NppStatus nppiMulScale\_8u\_C1IR** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.12 NppStatus nppiMulScale\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.18.2.13 NppStatus nppiMulScale\_8u\_C3IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.18.2.14 **NppStatus nppiMulScale\_8u\_C3R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.18.2.15 **NppStatus nppiMulScale\_8u\_C4IR** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.18.2.16 **NppStatus nppiMulScale\_8u\_C4R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image multiplication then scale by maximum value for pixel bit width.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.



*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.19 Sub

Pixel by pixel subtraction of two images.

### Functions

- `NppStatus nppiSub_8u_C1RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C1IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C3RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C3IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_AC4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_AC4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C4RSfs` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_8u_C4IRSfs` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSub_16u_C1RSfs` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*

- **NppStatus nppiSub\_32s\_C1IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32s\_C3RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32s\_C3IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32s\_C4RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32s\_C4IRSfs** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32sc\_C1RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32sc\_C1IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32sc\_C3RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSub\_32sc\_C3IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_AC4RSfs** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32sc\_AC4IRSfs** (const **Npp32sc** \*pSrc, int nSrcStep, **Npp32sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSub\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image subtraction.*
- **NppStatus nppiSub\_32f\_C1IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image subtraction.*
- **NppStatus nppiSub\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image subtraction.*
- **NppStatus nppiSub\_32f\_C3IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image subtraction.*
- **NppStatus nppiSub\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image subtraction.*
- **NppStatus nppiSub\_32f\_AC4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image subtraction.*
- **NppStatus nppiSub\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image subtraction.*
- **NppStatus nppiSub\_32f\_C4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image subtraction.*
- **NppStatus nppiSub\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*

- **NppStatus nppiSub\_32fc\_C1IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*
- **NppStatus nppiSub\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*
- **NppStatus nppiSub\_32fc\_C3IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*
- **NppStatus nppiSub\_32fc\_AC4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.*
- **NppStatus nppiSub\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.*
- **NppStatus nppiSub\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.*
- **NppStatus nppiSub\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.*

## 7.19.1 Detailed Description

Pixel by pixel subtraction of two images.

## 7.19.2 Function Documentation

### 7.19.2.1 NppStatus nppiSub\_16s\_AC4IRSfs (const Npp16s \*pSrc, int nSrcStep, Npp16s \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pSrcDst** In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.2 NppStatus npipiSub\_16s\_AC4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit signed short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.3 NppStatus npipiSub\_16s\_C1IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



#### 7.19.2.4 NppStatus nppiSub\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.5 NppStatus nppiSub\_16s\_C3IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.6 NppStatus nppiSub\_16s\_C3RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.7 **NppStatus nppiSub\_16s\_C4IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.8 **NppStatus nppiSub\_16s\_C4RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.9 **NppStatus nppiSub\_16sc\_AC4IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.10 **NppStatus nppiSub\_16sc\_AC4RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.11 **NppStatus nppiSub\_16sc\_C1IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.12 **NppStatus nppiSub\_16sc\_C1RSfs** (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.13 **NppStatus nppiSub\_16sc\_C3IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.14** `NppStatus nppiSub_16sc_C3RSfs (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, Npp16sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.15** `NppStatus nppiSub_16u_AC4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.16** `NppStatus nppiSub_16u_AC4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.17 **NppStatus nppiSub\_16u\_C1RSfs** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.18 **NppStatus nppiSub\_16u\_C1RSfs** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.19 NppStatus nppiSub\_16u\_C3IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.20 NppStatus nppiSub\_16u\_C3RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.21 `NppStatus nppiSub_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.22 `NppStatus nppiSub_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.23 `NppStatus nppiSub_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image subtraction.

#### Parameters:

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.24 **NppStatus nppiSub\_32f\_AC4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel with unmodified alpha image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.25 **NppStatus nppiSub\_32f\_C1IR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.26 `NppStatus nppiSub_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel image subtraction.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.19.2.27 `NppStatus nppiSub_32f_C3IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel in place image subtraction.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.19.2.28 `NppStatus nppiSub_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit floating point channel image subtraction.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.29 NppStatus nppiSub\_32f\_C4IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.30 NppStatus nppiSub\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.31 **NppStatus nppiSub\_32fc\_AC4IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.32 **NppStatus nppiSub\_32fc\_AC4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.33 **NppStatus nppiSub\_32fc\_C1IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.34 NppStatus npipiSub\_32fc\_C1R (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.35 NppStatus npipiSub\_32fc\_C3IR (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.36 **NppStatus nppiSub\_32fc\_C3R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.19.2.37 **NppStatus nppiSub\_32fc\_C4IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image subtraction.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.19.2.38 **NppStatus nppiSub\_32fc\_C4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image subtraction.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.39 NppStatus nppiSub\_32s\_C1IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.40 NppStatus nppiSub\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image subtraction. Subtract *pSrc1*'s pixels from corresponding pixels in *pSrc2*.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.41 NppStatus nppiSub\_32s\_C1RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.42 NppStatus nppiSub\_32s\_C3IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.43 NppStatus nppiSub\_32s\_C3RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.



**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.44 NppStatus nppiSub\_32s\_C4IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.45 NppStatus nppiSub\_32s\_C4RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 32-bit signed integer channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.46 NppStatus nppiSub\_32sc\_AC4IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.47 NppStatus nppiSub\_32sc\_AC4RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image subtraction, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.48 NppStatus nppiSub\_32sc\_C1IRSfs (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.49 NppStatus nppiSub\_32sc\_C1RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.19.2.50 NppStatus nppiSub\_32sc\_C3IRSfs (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.51** `NppStatus npplSub_32sc_C3RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.52** `NppStatus npplSub_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.53** `NppStatus nppiSub_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel with unmodified alpha image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.54** `NppStatus nppiSub_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.55** `NppStatus nppiSub_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.56 NppStatus nppiSub\_8u\_C3IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.19.2.57 NppStatus nppiSub\_8u\_C3RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.58 NppStatus nppiSub\_8u\_C4IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel in place image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.19.2.59 NppStatus nppiSub\_8u\_C4RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel image subtraction, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.20 Div

Pixel by pixel division of two images.

### Functions

- **NppStatus** **nppiDiv\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiDiv\_16u\_C1IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)



*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C3IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_AC4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16u\_C4IRSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C1IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C1RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C1IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C3RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_C3IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_AC4RSfs** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_16sc\_AC4IRSfs** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_32s\_C1RSfs** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.*

- `NppStatus nppiDiv_32s_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32s_C3RSfs` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32s_C3IRSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C1RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C1IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C3RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32sc_C3IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32sc_AC4RSfs` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32sc_AC4IRSfs` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

- `NppStatus nppiDiv_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image division.*
- `NppStatus nppiDiv_32f_C1IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image division.*
- `NppStatus nppiDiv_32f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image division.*
- `NppStatus nppiDiv_32f_C3IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image division.*
- `NppStatus nppiDiv_32f_AC4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha image division.*
- `NppStatus nppiDiv_32f_AC4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel with unmodified alpha in place image division.*
- `NppStatus nppiDiv_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel image division.*
- `NppStatus nppiDiv_32f_C4IR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel in place image division.*
- `NppStatus nppiDiv_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*
- `NppStatus nppiDiv_32fc_C1IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*
- `NppStatus nppiDiv_32fc_C3R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*
- `NppStatus nppiDiv_32fc_C3IR` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*
- `NppStatus nppiDiv_32fc_AC4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.*

- **NppStatus nppiDiv\_32fc\_AC4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.*

- **NppStatus nppiDiv\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.*

- **NppStatus nppiDiv\_32fc\_C4IR** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32fc** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.*

## 7.20.1 Detailed Description

Pixel by pixel division of two images.

## 7.20.2 Function Documentation

### 7.20.2.1 NppStatus nppiDiv\_16s\_AC4IRSfs (const Npp16s \*pSrc, int nSrcStep, Npp16s \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pSrcDst** In-Place Image Pointer.  
**nSrcDstStep** In-Place-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).  
**nScaleFactor** Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.2 NppStatus nppiDiv\_16s\_AC4RSfs (const Npp16s \*pSrc1, int nSrc1Step, const Npp16s \*pSrc2, int nSrc2Step, Npp16s \*pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

Four 16-bit signed short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

**pSrc1** Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.3 NppStatus nppiDiv\_16s\_C1IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.4 NppStatus nppiDiv\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.5 **NppStatus nppiDiv\_16s\_C3IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.6 **NppStatus nppiDiv\_16s\_C3RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.7 **NppStatus nppiDiv\_16s\_C4IRSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.8 **NppStatus nppiDiv\_16s\_C4RSfs** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.9 **NppStatus nppiDiv\_16sc\_AC4IRSfs** (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.20.2.10 NppStatus nppiDiv\_16sc\_AC4RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.11 NppStatus nppiDiv\_16sc\_C1IRSfs (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.12 NppStatus nppiDiv\_16sc\_C1RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.13 NppStatus nppiDiv\_16sc\_C3IRSfs (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.14 NppStatus nppiDiv\_16sc\_C3RSfs (const Npp16sc \* *pSrc1*, int *nSrc1Step*, const Npp16sc \* *pSrc2*, int *nSrc2Step*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short complex number (16-bit real, 16-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.15 NppStatus nppiDiv\_16u\_AC4IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.16 NppStatus nppiDiv\_16u\_AC4RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.17 NppStatus nppiDiv\_16u\_C1IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.18 NppStatus nppiDiv\_16u\_C1RSfs (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.19 NppStatus nppiDiv\_16u\_C3IRSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.20** `NppStatus nppiDiv_16u_C3RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.21** `NppStatus nppiDiv_16u_C4IRSfs (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.22** `NppStatus nppiDiv_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.23** `NppStatus nppiDiv_32f_AC4IR (const Npp32f * pSrc, int nSrcStep, Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.24** `NppStatus nppiDiv_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point channel with unmodified alpha image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.25 NppStatus nppiDiv\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.26 NppStatus nppiDiv\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.27 NppStatus nppiDiv\_32f\_C3IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.28 NppStatus nppiDiv\_32f\_C3R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.29 NppStatus nppiDiv\_32f\_C4IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.20.2.30 **NppStatus nppiDiv\_32f\_C4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point channel image division.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.31 **NppStatus nppiDiv\_32fc\_AC4IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.32 **NppStatus nppiDiv\_32fc\_AC4R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.33 NppStatus nppiDiv\_32fc\_C1IR (const Npp32fc \* pSrc, int nSrcStep, Npp32fc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.34 NppStatus nppiDiv\_32fc\_C1R (const Npp32fc \* pSrc1, int nSrc1Step, const Npp32fc \* pSrc2, int nSrc2Step, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.35 **NppStatus nppiDiv\_32fc\_C3IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.20.2.36 **NppStatus nppiDiv\_32fc\_C3R** (const Npp32fc \* *pSrc1*, int *nSrc1Step*, const Npp32fc \* *pSrc2*, int *nSrc2Step*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.20.2.37 **NppStatus nppiDiv\_32fc\_C4IR** (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel in place image division.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.38** `NppStatus nppiDiv_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit floating point complex number (32-bit real, 32-bit imaginary) channel image division.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.39** `NppStatus nppiDiv_32s_C1IRSfs (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.40** `NppStatus nppiDiv_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Note: This function is to be deprecated in future NPP releases, use the function above with a scale factor of 0 instead.

32-bit image division. Divide pixels in pSrc2 by pSrc1's pixels.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.41 NppStatus nppiDiv\_32s\_C1RSfs (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.42 NppStatus nppiDiv\_32s\_C3IRSfs (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.43** `NppStatus nppiDiv_32s_C3RSfs (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 32-bit signed integer channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.44** `NppStatus nppiDiv_32sc_AC4IRSfs (const Npp32sc * pSrc, int nSrcStep, Npp32sc * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.45** `NppStatus nppiDiv_32sc_AC4RSfs (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, Npp32sc * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel with unmodified alpha image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.46 NppStatus nppiDiv\_32sc\_C1RSfs (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.20.2.47 NppStatus nppiDiv\_32sc\_C1RSfs (const Npp32sc \* *pSrc1*, int *nSrc1Step*, const Npp32sc \* *pSrc2*, int *nSrc2Step*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.48 NppStatus nppiDiv\_32sc\_C3IRSfs (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.49 NppStatus nppiDiv\_32sc\_C3RSfs (const Npp32sc \* pSrc1, int nSrc1Step, const Npp32sc \* pSrc2, int nSrc2Step, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 32-bit signed integer complex number (32-bit real, 32-bit imaginary) channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.20.2.50 **NppStatus nppiDiv\_8u\_AC4IRSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.51 **NppStatus nppiDiv\_8u\_AC4RSfs** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel with unmodified alpha image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.20.2.52 **NppStatus nppiDiv\_8u\_C1IRSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.53** `NppStatus nppiDiv_8u_C1RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.54** `NppStatus nppiDiv_8u_C3IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.20.2.55** `NppStatus nppiDiv_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.56** `NppStatus nppiDiv_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.20.2.57** `NppStatus nppiDiv_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.21 Div\_Round

Pixel by pixel division of two images using result rounding modes.

### Functions

- **NppStatus nppiDiv\_Round\_8u\_C1RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C1IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C3RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C3IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_AC4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_AC4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C4RSfs** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_8u\_C4IRSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16u\_C1RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C3RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Three 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_AC4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C4RSfs** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16u\_C4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C1RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

*One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiDiv\_Round\_16s\_C3RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_C3IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Three 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_AC4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_C4RSfs** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiDiv\_Round\_16s\_C4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)  
*Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

## 7.21.1 Detailed Description

Pixel by pixel division of two images using result rounding modes.

## 7.21.2 Function Documentation

### 7.21.2.1 **NppStatus nppiDiv\_Round\_16s\_AC4IRSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, **NppRoundMode** rndMode, int nScaleFactor)

Four 16-bit signed short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pSrcDst** In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.21.2.2 NppStatus nppiDiv\_Round\_16s\_AC4RSfs (const Npp16s \* pSrc1, int nSrc1Step, const Npp16s \* pSrc2, int nSrc2Step, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)

Four 16-bit signed short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.21.2.3 NppStatus nppiDiv\_Round\_16s\_C1IRSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)

One 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).



*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.4 NppStatus nppiDiv\_Round\_16s\_C1RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)**

One 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.5 NppStatus nppiDiv\_Round\_16s\_C3IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)**

Three 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.21.2.6 NppStatus nppiDiv\_Round\_16s\_C3RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.21.2.7 NppStatus nppiDiv\_Round\_16s\_C4IRSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit signed short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.21.2.8 NppStatus nppiDiv\_Round\_16s\_C4RSfs (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit signed short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.21.2.9 NppStatus nppiDiv\_Round\_16u\_AC4IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.10 NppStatus nppiDiv\_Round\_16u\_AC4RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)**

Four 16-bit unsigned short channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.11 NppStatus nppiDiv\_Round\_16u\_C1IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)**

One 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.21.2.12 NppStatus nppiDiv\_Round\_16u\_C1RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

One 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.21.2.13 NppStatus nppiDiv\_Round\_16u\_C3IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.21.2.14 NppStatus nppiDiv\_Round\_16u\_C3RSfs (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Three 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.21.2.15 NppStatus nppiDiv\_Round\_16u\_C4IRSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.16** `NppStatus nppiDiv_Round_16u_C4RSfs (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 16-bit unsigned short channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.17** `NppStatus nppiDiv_Round_8u_AC4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.18** `NppStatus nppiDiv_Round_8u_AC4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.19** `NppStatus nppiDiv_Round_8u_C1IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

One 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.21.2.20 NppStatus nppiDiv\_Round\_8u\_C1RSfs (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)**

One 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.21.2.21 NppStatus nppiDiv\_Round\_8u\_C3IRSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, NppRoundMode *rndMode*, int *nScaleFactor*)**

Three 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.21.2.22 `NppStatus nppiDiv_Round_8u_C3RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Three 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.21.2.23 `NppStatus nppiDiv_Round_8u_C4IRSfs (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel in place image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.21.2.24** `NppStatus nppiDiv_Round_8u_C4RSfs (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode rndMode, int nScaleFactor)`

Four 8-bit unsigned char channel image division, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rndMode* Result Rounding mode to be used (NPP\_RND\_ZERO, NPP\_RND\_NEAR, or NP\_RND\_FINANCIAL)

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.22 Abs

Absolute value of each pixel value in an image.

### Functions

- `NppStatus nppiAbs_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit signed short channel image absolute value.*
- `NppStatus nppiAbs_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit signed short channel in place image absolute value.*
- `NppStatus nppiAbs_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit signed short channel image absolute value.*
- `NppStatus nppiAbs_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit signed short channel in place image absolute value.*
- `NppStatus nppiAbs_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel image absolute value with unmodified alpha.*
- `NppStatus nppiAbs_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel in place image absolute value with unmodified alpha.*
- `NppStatus nppiAbs_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel image absolute value.*
- `NppStatus nppiAbs_16s_C4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel in place image absolute value.*
- `NppStatus nppiAbs_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image absolute value.*
- `NppStatus nppiAbs_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image absolute value.*
- `NppStatus nppiAbs_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image absolute value.*
- `NppStatus nppiAbs_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel in place image absolute value.*
- `NppStatus nppiAbs_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit floating point channel image absolute value with unmodified alpha.*

- **NppStatus nppiAbs\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image absolute value with unmodified alpha.*
- **NppStatus nppiAbs\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel image absolute value.*
- **NppStatus nppiAbs\_32f\_C4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit floating point channel in place image absolute value.*

### 7.22.1 Detailed Description

Absolute value of each pixel value in an image.

### 7.22.2 Function Documentation

#### 7.22.2.1 NppStatus nppiAbs\_16s\_AC4IR (Npp16s \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel in place image absolute value with unmodified alpha.

##### Parameters:

**pSrcDst** In-Place Image Pointer.  
**nSrcDstStep** In-Place-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.22.2.2 NppStatus nppiAbs\_16s\_AC4R (const Npp16s \*pSrc, int nSrcStep, Npp16s \*pDst, int nDstStep, NppiSize oSizeROI)

Four 16-bit signed short channel image absolute value with unmodified alpha.

##### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.22.2.3 NppStatus nppiAbs\_16s\_C1IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.22.2.4 NppStatus nppiAbs\_16s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.22.2.5 NppStatus nppiAbs\_16s\_C3IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.6 NppStatus nppiAbs\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.7 NppStatus nppiAbs\_16s\_C4IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 16-bit signed short channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.8 NppStatus nppiAbs\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 16-bit signed short channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.9 NppStatus nppiAbs\_32f\_AC4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image absolute value with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.10 NppStatus nppiAbs\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image absolute value with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.11 NppStatus nppiAbs\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.22.2.12 NppStatus nppiAbs\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.13 NppStatus nppiAbs\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.14 NppStatus nppiAbs\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.15 NppStatus nppiAbs\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image absolute value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.22.2.16 NppStatus nppiAbs\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image absolute value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.23 AbsDiff

Pixel by pixel absolute difference between two images.

### Functions

- **NppStatus nppiAbsDiff\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel absolute difference of image1 minus image2.*
- **NppStatus nppiAbsDiff\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channels absolute difference of image1 minus image2.*
- **NppStatus nppiAbsDiff\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channels absolute difference of image1 minus image2.*
- **NppStatus nppiAbsDiff\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel absolute difference of image1 minus image2.*
- **NppStatus nppiAbsDiff\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel absolute difference of image1 minus image2.*

### 7.23.1 Detailed Description

Pixel by pixel absolute difference between two images.

### 7.23.2 Function Documentation

#### 7.23.2.1 **NppStatus nppiAbsDiff\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

One 16-bit unsigned short channel absolute difference of image1 minus image2.

#### Parameters:

- pSrc1** Source-Image Pointer.
- nSrc1Step** Source-Image Line Step.
- pSrc2** Source-Image Pointer.
- nSrc2Step** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.23.2.2** `NppStatus nppiAbsDiff_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

One 32-bit floating point channel absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.23.2.3** `NppStatus nppiAbsDiff_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

One 8-bit unsigned char channel absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.23.2.4** `NppStatus nppiAbsDiff_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channels absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.23.2.5** `NppStatus nppiAbsDiff_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channels absolute difference of image1 minus image2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.24 Sqr

Square each pixel in an image.

### Functions

- **NppStatus nppiSqr\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_AC4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_AC4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_8u\_C4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqr\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_AC4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16u\_C4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Four 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*One 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqr\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

*Three 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- `NppStatus nppiSqr_16s_AC4RSfs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_16s_AC4IRSfs` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_16s_C4RSfs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_16s_C4IRSfs` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- `NppStatus nppiSqr_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel image squared.*
- `NppStatus nppiSqr_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit floating point channel in place image squared.*
- `NppStatus nppiSqr_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel image squared.*
- `NppStatus nppiSqr_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit floating point channel in place image squared.*
- `NppStatus nppiSqr_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel image squared with unmodified alpha.*
- `NppStatus nppiSqr_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel in place image squared with unmodified alpha.*
- `NppStatus nppiSqr_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel image squared.*
- `NppStatus nppiSqr_32f_C4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit floating point channel in place image squared.*

### 7.24.1 Detailed Description

Square each pixel in an image.



## 7.24.2 Function Documentation

### 7.24.2.1 NppStatus nppiSqr\_16s\_AC4IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.2 NppStatus nppiSqr\_16s\_AC4RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.3 NppStatus nppiSqr\_16s\_C1IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.24.2.4 NppStatus nppiSqr\_16s\_C1RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.5 NppStatus nppiSqr\_16s\_C3IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.6 NppStatus nppiSqr\_16s\_C3RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.24.2.7 NppStatus nppiSqr\_16s\_C4IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.8 NppStatus nppiSqr\_16s\_C4RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.9 NppStatus nppiSqr\_16u\_AC4IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.24.2.10 **NppStatus nppiSqr\_16u\_AC4RSfs** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit unsigned short channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.11 **NppStatus nppiSqr\_16u\_C1IRSfs** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.12 **NppStatus nppiSqr\_16u\_C1RSfs** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.13 NppStatus nppiSqr\_16u\_C3IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.14 NppStatus nppiSqr\_16u\_C3RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit unsigned short channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.15 NppStatus nppiSqr\_16u\_C4IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.16 NppStatus nppiSqr\_16u\_C4RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel image squared, scale by  $2^{-(nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.17 NppStatus nppiSqr\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel in place image squared with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.18 NppStatus nppiSqr\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit floating point channel image squared with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.19 NppStatus nppiSqr\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.20 NppStatus nppiSqr\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image squared.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.21 NppStatus nppiSqr\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.22 NppStatus nppiSqr\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image squared.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.23 NppStatus nppiSqr\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image squared.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.24 NppStatus nppiSqr\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image squared.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.24.2.25 NppStatus nppiSqr\_8u\_AC4IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel in place image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.26 NppStatus nppiSqr\_8u\_AC4RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image squared with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.24.2.27 NppStatus nppiSqr\_8u\_C1IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.24.2.28 **NppStatus nppiSqr\_8u\_C1RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.29 **NppStatus nppiSqr\_8u\_C3IRSfs** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.24.2.30 **NppStatus nppiSqr\_8u\_C3RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.31 NppStatus nppiSqr\_8u\_C4IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel in place image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.24.2.32 NppStatus nppiSqr\_8u\_C4RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel image squared, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.25 Sqrt

Pixel by pixel square root of each pixel in an image.

### Functions

- **NppStatus** **nppiSqrt\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_AC4RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_8u\_AC4IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiSqrt\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiSqrt\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16u\_AC4RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16u\_AC4IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_AC4RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_16s\_AC4IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Four 16-bit signed short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiSqrt\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image square root.*
- **NppStatus nppiSqrt\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image square root.*

- [NppStatus nppiSqrt\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Three 32-bit floating point channel image square root.*

- [NppStatus nppiSqrt\\_32f\\_C3IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Three 32-bit floating point channel in place image square root.*

- [NppStatus nppiSqrt\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel image square root with unmodified alpha.*

- [NppStatus nppiSqrt\\_32f\\_AC4IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel in place image square root with unmodified alpha.*

- [NppStatus nppiSqrt\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel image square root.*

- [NppStatus nppiSqrt\\_32f\\_C4IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Four 32-bit floating point channel in place image square root.*

### 7.25.1 Detailed Description

Pixel by pixel square root of each pixel in an image.

### 7.25.2 Function Documentation

#### 7.25.2.1 [NppStatus nppiSqrt\\_16s\\_AC4IRSfs](#) ([Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, int nScaleFactor)

Four 16-bit signed short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.25.2.2 NppStatus nppiSqrt\_16s\_AC4RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 16-bit signed short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.3 NppStatus nppiSqrt\_16s\_C1IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.4 NppStatus nppiSqrt\_16s\_C1RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.5 NppStatus nppiSqrt\_16s\_C3IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.6 NppStatus nppiSqrt\_16s\_C3RSfs (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 16-bit signed short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.7 NppStatus nppiSqrt\_16u\_AC4IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.



*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.8 NppStatus nppiSqrt\_16u\_AC4RSfs (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Four 16-bit unsigned short channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.9 NppStatus nppiSqrt\_16u\_C1IRSfs (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

One 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.10 **NppStatus nppiSqrt\_16u\_C1RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.11 **NppStatus nppiSqrt\_16u\_C3IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.12 **NppStatus nppiSqrt\_16u\_C3RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 16-bit unsigned short channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.13 NppStatus nppiSqrt\_32f\_AC4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image square root with unmodified alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.14 NppStatus nppiSqrt\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image square root with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.15 NppStatus nppiSqrt\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.16 NppStatus nppiSqrt\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit floating point channel image square root.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.17 NppStatus nppiSqrt\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.18 NppStatus nppiSqrt\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image square root.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.19 NppStatus nppiSqrt\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel in place image square root.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.20 NppStatus nppiSqrt\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit floating point channel image square root.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.21 NppStatus nppiSqrt\_8u\_AC4IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Four 8-bit unsigned char channel in place image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.25.2.22 **NppStatus nppiSqrt\_8u\_AC4RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Four 8-bit unsigned char channel image square root with unmodified alpha, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.25.2.23 **NppStatus nppiSqrt\_8u\_C1IRSfs** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.25.2.24 **NppStatus nppiSqrt\_8u\_C1RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.25 NppStatus nppiSqrt\_8u\_C3IRSfs (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel in place image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.25.2.26 NppStatus nppiSqrt\_8u\_C3RSfs (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, int nScaleFactor)**

Three 8-bit unsigned char channel image square root, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.26 Ln

Pixel by pixel natural logarithm of each pixel in an image.

### Functions

- **NppStatus nppiLn\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*



- **NppStatus nppiLn\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiLn\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image natural logarithm.*
- **NppStatus nppiLn\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image natural logarithm.*
- **NppStatus nppiLn\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image natural logarithm.*
- **NppStatus nppiLn\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image natural logarithm.*

## 7.26.1 Detailed Description

Pixel by pixel natural logarithm of each pixel in an image.

## 7.26.2 Function Documentation

### 7.26.2.1 **NppStatus nppiLn\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

**pSrcDst** In-Place Image Pointer.  
**nSrcDstStep** In-Place-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).  
**nScaleFactor** Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.2 NppStatus nppiLn\_16s\_C1RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.3 NppStatus nppiLn\_16s\_C3IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.4 NppStatus nppiLn\_16s\_C3RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.5 NppStatus nppiLn\_16u\_C1IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.26.2.6 NppStatus nppiLn\_16u\_C1RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.26.2.7 NppStatus nppiLn\_16u\_C3IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.26.2.8 NppStatus nppiLn\_16u\_C3RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.9 NppStatus nppiLn\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image natural logarithm.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.26.2.10 NppStatus nppiLn\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image natural logarithm.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.26.2.11 NppStatus nppiLn\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image natural logarithm.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.26.2.12 NppStatus nppiLn\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image natural logarithm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.26.2.13 NppStatus nppiLn\_8u\_C1IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.26.2.14 **NppStatus nppiLn\_8u\_C1RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.26.2.15 **NppStatus nppiLn\_8u\_C3IRSfs** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel in place image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.26.2.16 **NppStatus nppiLn\_8u\_C3RSfs** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 8-bit unsigned char channel image natural logarithm, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

## 7.27 Exp

Exponential value of each pixel in an image.

### Functions

- **NppStatus** **nppiExp\_8u\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_8u\_C1IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_8u\_C3RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_8u\_C3IRSfs** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_16u\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_16u\_C1IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_16u\_C3RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_16u\_C3IRSfs** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus** **nppiExp\_16s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*

- **NppStatus nppiExp\_16s\_C1IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*One 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C3RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_16s\_C3IRSfs** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, int nScaleFactor)  
*Three 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.*
- **NppStatus nppiExp\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel image exponential.*
- **NppStatus nppiExp\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit floating point channel in place image exponential.*
- **NppStatus nppiExp\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel image exponential.*
- **NppStatus nppiExp\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit floating point channel in place image exponential.*

## 7.27.1 Detailed Description

Exponential value of each pixel in an image.

## 7.27.2 Function Documentation

### 7.27.2.1 NppStatus nppiExp\_16s\_C1IRSfs (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, int nScaleFactor)

One 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

**pSrcDst** In-Place Image Pointer.  
**nSrcDstStep** In-Place-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).  
**nScaleFactor** Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes



### 7.27.2.2 NppStatus nppiExp\_16s\_C1RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.3 NppStatus nppiExp\_16s\_C3IRSfs (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.4 NppStatus nppiExp\_16s\_C3RSfs (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit signed short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.5 NppStatus nppiExp\_16u\_C1IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.27.2.6 NppStatus nppiExp\_16u\_C1RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

One 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.27.2.7 NppStatus nppiExp\_16u\_C3IRSfs (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.27.2.8 NppStatus nppiExp\_16u\_C3RSfs (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)

Three 16-bit unsigned short channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.9 NppStatus nppiExp\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel in place image exponential.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.27.2.10 NppStatus nppiExp\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit floating point channel image exponential.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.11 NppStatus nppiExp\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel in place image exponential.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.27.2.12 NppStatus nppiExp\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 32-bit floating point channel image exponential.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.27.2.13 NppStatus nppiExp\_8u\_C1IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.27.2.14 NppStatus nppiExp\_8u\_C1RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

One 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.15 NppStatus nppiExp\_8u\_C3IRSfs (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 8-bit unsigned char channel in place image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.27.2.16 NppStatus nppiExp\_8u\_C3RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, int *nScaleFactor*)**

Three 8-bit unsigned char channel image exponential, scale by  $2^{(-nScaleFactor)}$ , then clamp to saturated value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.28 Logical Operations

### Modules

- [AndC](#)

*Pixel by pixel logical and of an image with a constant.*

- [OrC](#)

*Pixel by pixel logical or of an image with a constant.*

- [XorC](#)

*Pixel by pixel logical exclusive or of an image with a constant.*

- [RShiftC](#)

*Pixel by pixel right shift of an image by a constant value.*

- [LShiftC](#)

*Pixel by pixel left shift of an image by a constant value.*

- [And](#)

*Pixel by pixel logical and of images.*

- [Or](#)

*Pixel by pixel logical or of images.*

- [Xor](#)

*Pixel by pixel logical exclusive or of images.*

- [Not](#)

*Pixel by pixel logical not of image.*

## 7.29 AndC

Pixel by pixel logical and of an image with a constant.

### Functions

- **NppStatus nppiAndC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C3IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical and with constant.*
- **NppStatus nppiAndC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.*
- **NppStatus nppiAndC\_8u\_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.*
- **NppStatus nppiAndC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and with constant.*
- **NppStatus nppiAndC\_8u\_C4IR** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with constant.*
- **NppStatus nppiAndC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical and with constant.*
- **NppStatus nppiAndC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical and with constant.*
- **NppStatus nppiAndC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical and with constant.*

- `NppStatus nppiAndC_16u_C3IR` (const `Npp16u` `aConstants[3]`, `Npp16u` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*Three 16-bit unsigned short channel in place image logical and with constant.*
- `NppStatus nppiAndC_16u_AC4R` (const `Npp16u` `*pSrc1`, int `nSrc1Step`, const `Npp16u` `aConstants[3]`, `Npp16u` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_16u_AC4IR` (const `Npp16u` `aConstants[3]`, `Npp16u` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_16u_C4R` (const `Npp16u` `*pSrc1`, int `nSrc1Step`, const `Npp16u` `aConstants[4]`, `Npp16u` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four 16-bit unsigned short channel image logical and with constant.*
- `NppStatus nppiAndC_16u_C4IR` (const `Npp16u` `aConstants[4]`, `Npp16u` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*Four 16-bit unsigned short channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_C1R` (const `Npp32s` `*pSrc1`, int `nSrc1Step`, const `Npp32s` `nConstant`, `Npp32s` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*One 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C1IR` (const `Npp32s` `nConstant`, `Npp32s` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*One 32-bit signed integer channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_C3R` (const `Npp32s` `*pSrc1`, int `nSrc1Step`, const `Npp32s` `aConstants[3]`, `Npp32s` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Three 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C3IR` (const `Npp32s` `aConstants[3]`, `Npp32s` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*Three 32-bit signed integer channel in place image logical and with constant.*
- `NppStatus nppiAndC_32s_AC4R` (const `Npp32s` `*pSrc1`, int `nSrc1Step`, const `Npp32s` `aConstants[3]`, `Npp32s` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four 32-bit signed integer channel image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_32s_AC4IR` (const `Npp32s` `aConstants[3]`, `Npp32s` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.*
- `NppStatus nppiAndC_32s_C4R` (const `Npp32s` `*pSrc1`, int `nSrc1Step`, const `Npp32s` `aConstants[4]`, `Npp32s` `*pDst`, int `nDstStep`, `NppiSize` `oSizeROI`)  
*Four 32-bit signed integer channel image logical and with constant.*
- `NppStatus nppiAndC_32s_C4IR` (const `Npp32s` `aConstants[4]`, `Npp32s` `*pSrcDst`, int `nSrcDstStep`, `NppiSize` `oSizeROI`)  
*Four 32-bit signed integer channel in place image logical and with constant.*



### 7.29.1 Detailed Description

Pixel by pixel logical and of an image with a constant.

### 7.29.2 Function Documentation

#### 7.29.2.1 `NppStatus nppiAndC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and with constant with unmodified alpha.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.29.2.2 `NppStatus nppiAndC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with constant with unmodified alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.29.2.3 `NppStatus nppiAndC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical and with constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.4 NppStatus nppiAndC\_16u\_C1R (const Npp16u \*pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.5 NppStatus nppiAndC\_16u\_C3IR (const Npp16u aConstants[3], Npp16u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.6 NppStatus nppiAndC\_16u\_C3R (const Npp16u \*pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.7 NppStatus nppiAndC\_16u\_C4IR (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical and with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.29.2.8 NppStatus nppiAndC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.9 NppStatus nppiAndC\_32s\_AC4IR (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical and with constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.10 NppStatus nppiAndC\_32s\_AC4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical and with constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.11 NppStatus nppiAndC\_32s\_C1IR (const Npp32s *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical and with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.12 NppStatus nppiAndC\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical and with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.13 NppStatus nppiAndC\_32s\_C3IR (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical and with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.14 NppStatus nppiAndC\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical and with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.15 `NppStatus nppiAndC_32s_C4IR (const Npp32s aConstants[4], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical and with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.29.2.16 `NppStatus nppiAndC_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s aConstants[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical and with constant.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.29.2.17 `NppStatus nppiAndC_8u_AC4IR (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and with constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.29.2.18 NppStatus nppiAndC\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical and with constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.19 NppStatus nppiAndC\_8u\_C1IR (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical and with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.20 NppStatus nppiAndC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical and with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.29.2.21 `NppStatus nppiAndC_8u_C3IR (const Npp8u aConstants[3], Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical and with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.29.2.22 `NppStatus nppiAndC_8u_C3R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u aConstants[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical and with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.29.2.23 `NppStatus nppiAndC_8u_C4IR (const Npp8u aConstants[4], Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.29.2.24** `NppStatus nppiAndC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.30 OrC

Pixel by pixel logical or of an image with a constant.

### Functions

- **NppStatus nppiOrC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C3IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.*
- **NppStatus nppiOrC\_8u\_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.*
- **NppStatus nppiOrC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or with constant.*
- **NppStatus nppiOrC\_8u\_C4IR** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical or with constant.*
- **NppStatus nppiOrC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical or with constant.*

- `NppStatus nppiOrC_16u_C3IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image logical or with constant.*
- `NppStatus nppiOrC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_16u_AC4IR` (const `Npp16u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with constant.*
- `NppStatus nppiOrC_16u_C4IR` (const `Npp16u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C1IR` (const `Npp32s` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C3IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical or with constant.*
- `NppStatus nppiOrC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_32s_AC4IR` (const `Npp32s` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.*
- `NppStatus nppiOrC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` aConstants[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with constant.*
- `NppStatus nppiOrC_32s_C4IR` (const `Npp32s` aConstants[4], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with constant.*

### 7.30.1 Detailed Description

Pixel by pixel logical or of an image with a constant.

### 7.30.2 Function Documentation

#### 7.30.2.1 `NppStatus nppiOrC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with constant with unmodified alpha.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.30.2.2 `NppStatus nppiOrC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with constant with unmodified alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.30.2.3 `NppStatus nppiOrC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical or with constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.4 NppStatus nppiOrC\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *nConstant*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.5 NppStatus nppiOrC\_16u\_C3IR (const Npp16u *aConstants*[3], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.6 NppStatus nppiOrC\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.7 NppStatus nppiOrC\_16u\_C4IR (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.30.2.8 NppStatus nppiOrC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.30.2.9 NppStatus nppiOrC\_32s\_AC4IR (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical or with constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.10 NppStatus nppiOrC\_32s\_AC4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical or with constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.11 NppStatus nppiOrC\_32s\_C1IR (const Npp32s *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical or with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.12 NppStatus nppiOrC\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.30.2.13 NppStatus nppiOrC\_32s\_C3IR (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.30.2.14 NppStatus nppiOrC\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes



### 7.30.2.15 NppStatus nppiOrC\_32s\_C4IR (const Npp32s *aConstants*[4], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.16 NppStatus nppiOrC\_32s\_C4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[4], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.17 NppStatus nppiOrC\_8u\_AC4IR (const Npp8u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical or with constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.18 **NppStatus nppiOrC\_8u\_AC4R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical or with constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.19 **NppStatus nppiOrC\_8u\_C1IR** (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical or with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.20 **NppStatus nppiOrC\_8u\_C1R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.30.2.21 NppStatus nppiOrC\_8u\_C3IR (const Npp8u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.30.2.22 NppStatus nppiOrC\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image logical or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.30.2.23 NppStatus nppiOrC\_8u\_C4IR (const Npp8u *aConstants*[4], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.30.2.24 NppStatus nppiOrC\_8u\_C4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[4], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

## 7.31 XorC

Pixel by pixel logical exclusive or of an image with a constant.

### Functions

- **NppStatus nppiXorC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C1IR** (const **Npp8u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C3IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_8u\_AC4IR** (const **Npp8u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_8u\_C4IR** (const **Npp8u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C1IR** (const **Npp16u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical exclusive or with constant.*

- **NppStatus nppiXorC\_16u\_C3IR** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_16u\_AC4IR** (const **Npp16u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** aConstants[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_16u\_C4IR** (const **Npp16u** aConstants[4], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** nConstant, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_32s\_C1IR** (const **Npp32s** nConstant, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_32s\_C3R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** aConstants[3], **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_32s\_C3IR** (const **Npp32s** aConstants[3], **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel in place image logical exclusive or with constant.*
- **NppStatus nppiXorC\_32s\_AC4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** aConstants[3], **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_32s\_AC4IR** (const **Npp32s** aConstants[3], **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.*
- **NppStatus nppiXorC\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** aConstants[4], **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with constant.*
- **NppStatus nppiXorC\_32s\_C4IR** (const **Npp32s** aConstants[4], **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with constant.*

### 7.31.1 Detailed Description

Pixel by pixel logical exclusive or of an image with a constant.

### 7.31.2 Function Documentation

#### 7.31.2.1 `NppStatus nppiXorC_16u_AC4IR (const Npp16u aConstants[3], Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with constant with unmodified alpha.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.31.2.2 `NppStatus nppiXorC_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with constant with unmodified alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.31.2.3 `NppStatus nppiXorC_16u_C1IR (const Npp16u nConstant, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical exclusive or with constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.4 NppStatus nppiXorC\_16u\_C1R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u nConstant, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.5 NppStatus nppiXorC\_16u\_C3IR (const Npp16u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.6 NppStatus nppiXorC\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.



*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.7 NppStatus nppiXorC\_16u\_C4IR (const Npp16u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image logical exclusive or with constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.31.2.8 NppStatus nppiXorC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.9 **NppStatus nppiXorC\_32s\_AC4IR** (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical exclusive or with constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.10 **NppStatus nppiXorC\_32s\_AC4R** (const Npp32s \* *pSrcI*, int *nSrcIStep*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical exclusive or with constant with unmodified alpha.

#### Parameters:

*pSrcI* Source-Image Pointer.

*nSrcIStep* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.11 **NppStatus nppiXorC\_32s\_C1IR** (const Npp32s *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical exclusive or with constant.

#### Parameters:

*nConstant* Constant.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.12 NppStatus nppiXorC\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical exclusive or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.13 NppStatus nppiXorC\_32s\_C3IR (const Npp32s *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical exclusive or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.14 NppStatus nppiXorC\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical exclusive or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.15 `NppStatus nppiXorC_32s_C4IR (const Npp32s aConstants[4], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical exclusive or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.16 `NppStatus nppiXorC_32s_C4R (const Npp32s * pSrcI, int nSrcIStep, const Npp32s aConstants[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical exclusive or with constant.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).  
*nSrcIStep* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.17 `NppStatus nppiXorC_8u_AC4IR (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or with constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.18 NppStatus nppiXorC\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical exclusive or with constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.19 NppStatus nppiXorC\_8u\_C1IR (const Npp8u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical exclusive or with constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.20 NppStatus nppiXorC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical exclusive or with constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.31.2.21 `NppStatus nppiXorC_8u_C3IR (const Npp8u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image logical exclusive or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.22 `NppStatus nppiXorC_8u_C3R (const Npp8u * pSrcI, int nSrcIStep, const Npp8u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical exclusive or with constant.

#### Parameters:

*pSrcI* [Source-Image Pointer](#).  
*nSrcIStep* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.31.2.23 `NppStatus nppiXorC_8u_C4IR (const Npp8u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or with constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.31.2.24** `NppStatus nppiXorC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or with constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.32 RShiftC

Pixel by pixel right shift of an image by a constant value.

### Functions

- **NppStatus nppiRShiftC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image right shift by constant.*
- **NppStatus nppiRShiftC\_8u\_C1IR** (const **Npp32u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image right shift by constant.*
- **NppStatus nppiRShiftC\_8u\_C3IR** (const **Npp32u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_8u\_AC4IR** (const **Npp32u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image right shift by constant.*
- **NppStatus nppiRShiftC\_8u\_C4IR** (const **Npp32u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_8s\_C1R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit signed char channel image right shift by constant.*
- **NppStatus nppiRShiftC\_8s\_C1IR** (const **Npp32u** nConstant, **Npp8s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit signed char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_8s\_C3R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit signed char channel image right shift by constant.*



- **NppStatus nppiRShiftC\_8s\_C3IR** (const **Npp32u** aConstants[3], **Npp8s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit signed char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_8s\_AC4R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_8s\_AC4IR** (const **Npp32u** aConstants[3], **Npp8s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_8s\_C4R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel image right shift by constant.*
- **NppStatus nppiRShiftC\_8s\_C4IR** (const **Npp32u** aConstants[4], **Npp8s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit signed char channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C1IR** (const **Npp32u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C3IR** (const **Npp32u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_16u\_AC4IR** (const **Npp32u** aConstants[3], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.*
- **NppStatus nppiRShiftC\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image right shift by constant.*
- **NppStatus nppiRShiftC\_16u\_C4IR** (const **Npp32u** aConstants[4], **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image right shift by constant.*

- `NppStatus nppiRShiftC_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit signed short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C1IR` (const `Npp32u` nConstant, `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit signed short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C3R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit signed short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C3IR` (const `Npp32u` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit signed short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_16s_AC4R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_16s_AC4IR` (const `Npp32u` aConstants[3], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.*
- `NppStatus nppiRShiftC_16s_C4R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel image right shift by constant.*
- `NppStatus nppiRShiftC_16s_C4IR` (const `Npp32u` aConstants[4], `Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit signed short channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C1IR` (const `Npp32u` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image right shift by constant.*
- `NppStatus nppiRShiftC_32s_C3IR` (const `Npp32u` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image right shift by constant.*
- `NppStatus nppiRShiftC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four 32-bit signed integer channel image right shift by constant with unmodified alpha.*

- **NppStatus nppiRShiftC\_32s\_AC4IR** (const **Npp32u** aConstants[3], **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.*

- **NppStatus nppiRShiftC\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 32-bit signed integer channel image right shift by constant.*

- **NppStatus nppiRShiftC\_32s\_C4IR** (const **Npp32u** aConstants[4], **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 32-bit signed integer channel in place image right shift by constant.*

### 7.32.1 Detailed Description

Pixel by pixel right shift of an image by a constant value.

### 7.32.2 Function Documentation

#### 7.32.2.1 **NppStatus nppiRShiftC\_16s\_AC4IR** (const **Npp32u** aConstants[3], **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit signed short channel in place image right shift by constant with unmodified alpha.

##### Parameters:

**aConstants** fixed size array of constant values, one per channel.

**pSrcDst** In-Place Image Pointer.

**nSrcDstStep** In-Place-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.32.2.2 **NppStatus nppiRShiftC\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

Four 16-bit signed short channel image right shift by constant with unmodified alpha.

##### Parameters:

**pSrc1** Source-Image Pointer.

**nSrc1Step** Source-Image Line Step.

**aConstants** fixed size array of constant values, one per channel.

**pDst** Destination-Image Pointer.

**nDstStep** Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.3 NppStatus nppiRShiftC\_16s\_C1IR (const Npp32u *nConstant*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 16-bit signed short channel in place image right shift by constant.

**Parameters:**

*nConstant* Constant.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.4 NppStatus nppiRShiftC\_16s\_C1R (const Npp16s \* *pSrcI*, int *nSrcIStep*, const Npp32u *nConstant*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrcI* [Source-Image Pointer](#).

*nSrcIStep* [Source-Image Line Step](#).

*nConstant* Constant.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.5 NppStatus nppiRShiftC\_16s\_C3IR (const Npp32u *aConstants*[3], Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 16-bit signed short channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.6 NppStatus nppiRShiftC\_16s\_C3R (const Npp16s \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.7 NppStatus nppiRShiftC\_16s\_C4IR (const Npp32u aConstants[4], Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit signed short channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.8 NppStatus nppiRShiftC\_16s\_C4R (const Npp16s \* pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit signed short channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.9 NppStatus nppiRShiftC\_16u\_AC4IR (const Npp32u aConstants[3], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image right shift by constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.10 NppStatus nppiRShiftC\_16u\_AC4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.11 NppStatus nppiRShiftC\_16u\_C1IR (const Npp32u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image right shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.12 NppStatus nppiRShiftC\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.13 NppStatus nppiRShiftC\_16u\_C3IR (const Npp32u *aConstants*[3], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image right shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.14 NppStatus nppiRShiftC\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.15 NppStatus nppiRShiftC\_16u\_C4IR (const Npp32u *aConstants*[4], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image right shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.16 NppStatus nppiRShiftC\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[4], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes



### 7.32.2.17 NppStatus nppiRShiftC\_32s\_AC4IR (const Npp32u *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image right shift by constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.18 NppStatus nppiRShiftC\_32s\_AC4R (const Npp32s \* *pSrcI*, int *nSrcIStep*, const Npp32u *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image right shift by constant with unmodified alpha.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.19 NppStatus nppiRShiftC\_32s\_C1IR (const Npp32u *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image right shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.20 **NppStatus nppiRShiftC\_32s\_C1R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.21 **NppStatus nppiRShiftC\_32s\_C3IR** (const Npp32u *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image right shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.22 **NppStatus nppiRShiftC\_32s\_C3R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.23 NppStatus nppiRShiftC\_32s\_C4IR (const Npp32u *aConstants*[4], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image right shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.32.2.24 NppStatus nppiRShiftC\_32s\_C4R (const Npp32s \* *pSrcI*, int *nSrcIStep*, const Npp32u *aConstants*[4], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image right shift by constant.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.32.2.25 NppStatus nppiRShiftC\_8s\_AC4IR (const Npp32u *aConstants*[3], Npp8s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit signed char channel in place image right shift by constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.32.2.26 **NppStatus nppiRShiftC\_8s\_AC4R** (const Npp8s \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit signed char channel image right shift by constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.27 **NppStatus nppiRShiftC\_8s\_C11R** (const Npp32u *nConstant*, Npp8s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit signed char channel in place image right shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.28 **NppStatus nppiRShiftC\_8s\_C1R** (const Npp8s \* *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit signed char channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.29 NppStatus nppiRShiftC\_8s\_C3IR (const Npp32u *aConstants*[3], Npp8s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit signed char channel in place image right shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.32.2.30 NppStatus nppiRShiftC\_8s\_C3R (const Npp8s \* *pSrcI*, int *nSrcIStep*, const Npp32u *aConstants*[3], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit signed char channel image right shift by constant.

#### Parameters:

*pSrcI* Source-Image Pointer.  
*nSrcIStep* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.32.2.31 NppStatus nppiRShiftC\_8s\_C4IR (const Npp32u *aConstants*[4], Npp8s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit signed char channel in place image right shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.32** `NppStatus nppiRShiftC_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit signed char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.33** `NppStatus nppiRShiftC_8u_AC4IR (const Npp32u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image right shift by constant with unmodified alpha.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.34** `NppStatus nppiRShiftC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image right shift by constant with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.32.2.35 NppStatus nppiRShiftC\_8u\_C1IR (const Npp32u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image right shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.32.2.36 NppStatus nppiRShiftC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image right shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.32.2.37 NppStatus nppiRShiftC\_8u\_C3IR (const Npp32u *aConstants*[3], Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image right shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.32.2.38** `NppStatus nppiRShiftC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.39** `NppStatus nppiRShiftC_8u_C4IR (const Npp32u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image right shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.32.2.40** `NppStatus nppiRShiftC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image right shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



## 7.33 LShiftC

Pixel by pixel left shift of an image by a constant value.

### Functions

- **NppStatus nppiLShiftC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C1IR** (const **Npp32u** nConstant, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C3IR** (const **Npp32u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.*
- **NppStatus nppiLShiftC\_8u\_AC4IR** (const **Npp32u** aConstants[3], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.*
- **NppStatus nppiLShiftC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image left shift by constant.*
- **NppStatus nppiLShiftC\_8u\_C4IR** (const **Npp32u** aConstants[4], **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** nConstant, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image left shift by constant.*
- **NppStatus nppiLShiftC\_16u\_C1IR** (const **Npp32u** nConstant, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image left shift by constant.*
- **NppStatus nppiLShiftC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp32u** aConstants[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image left shift by constant.*

- `NppStatus nppiLShiftC_16u_C3IR` (const `Npp32u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_16u_AC4IR` (const `Npp32u` aConstants[3], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image left shift by constant.*
- `NppStatus nppiLShiftC_16u_C4IR` (const `Npp32u` aConstants[4], `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` nConstant, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C1IR` (const `Npp32u` nConstant, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C3IR` (const `Npp32u` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image left shift by constant.*
- `NppStatus nppiLShiftC_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_32s_AC4IR` (const `Npp32u` aConstants[3], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.*
- `NppStatus nppiLShiftC_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32u` aConstants[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image left shift by constant.*
- `NppStatus nppiLShiftC_32s_C4IR` (const `Npp32u` aConstants[4], `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image left shift by constant.*

### 7.33.1 Detailed Description

Pixel by pixel left shift of an image by a constant value.

### 7.33.2 Function Documentation

#### 7.33.2.1 NppStatus nppiLShiftC\_16u\_AC4IR (const Npp32u *aConstants*[3], Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image left shift by constant with unmodified alpha.

##### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.33.2.2 NppStatus nppiLShiftC\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image left shift by constant with unmodified alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.33.2.3 NppStatus nppiLShiftC\_16u\_C1IR (const Npp32u *nConstant*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel in place image left shift by constant.

##### Parameters:

*nConstant* Constant.  
*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.4 NppStatus nppiLShiftC\_16u\_C1R (const Npp16u \*pSrc1, int nSrc1Step, const Npp32u nConstant, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)**

One 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nConstant* Constant

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.5 NppStatus nppiLShiftC\_16u\_C3IR (const Npp32u aConstants[3], Npp16u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel in place image left shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.6 NppStatus nppiLShiftC\_16u\_C3R (const Npp16u \*pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)**

Three 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.7 NppStatus nppiLShiftC\_16u\_C4IR (const Npp32u aConstants[4], Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel in place image left shift by constant.

**Parameters:**

*aConstants* fixed size array of constant values, one per channel.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.33.2.8 NppStatus nppiLShiftC\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 16-bit unsigned short channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.9 NppStatus nppiLShiftC\_32s\_AC4IR (const Npp32u *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image left shift by constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.10 NppStatus nppiLShiftC\_32s\_AC4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image left shift by constant with unmodified alpha.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.11 NppStatus nppiLShiftC\_32s\_C1IR (const Npp32u *nConstant*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image left shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.12 NppStatus nppiLShiftC\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image left shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.13 NppStatus nppiLShiftC\_32s\_C3IR (const Npp32u *aConstants*[3], Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image left shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.14 NppStatus nppiLShiftC\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image left shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.15 `NppStatus nppiLShiftC_32s_C4IR (const Npp32u aConstants[4], Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image left shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.16 `NppStatus nppiLShiftC_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image left shift by constant.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.17 `NppStatus nppiLShiftC_8u_AC4IR (const Npp32u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image left shift by constant with unmodified alpha.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.33.2.18 NppStatus nppiLShiftC\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp32u *aConstants*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image left shift by constant with unmodified alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.19 NppStatus nppiLShiftC\_8u\_C1IR (const Npp32u *nConstant*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image left shift by constant.

#### Parameters:

*nConstant* Constant.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.20 NppStatus nppiLShiftC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp32u *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image left shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nConstant* Constant.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.33.2.21 `NppStatus nppiLShiftC_8u_C3IR (const Npp32u aConstants[3], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel in place image left shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.22 `NppStatus nppiLShiftC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image left shift by constant.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*aConstants* fixed size array of constant values, one per channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.33.2.23 `NppStatus nppiLShiftC_8u_C4IR (const Npp32u aConstants[4], Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image left shift by constant.

#### Parameters:

*aConstants* fixed size array of constant values, one per channel.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.33.2.24** `NppStatus nppiLShiftC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp32u aConstants[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image left shift by constant.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*aConstants* fixed size array of constant values, one per channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.34 And

Pixel by pixel logical and of images.

### Functions

- **NppStatus** **nppiAnd\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical and.*
- **NppStatus** **nppiAnd\_8u\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical and.*
- **NppStatus** **nppiAnd\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical and.*
- **NppStatus** **nppiAnd\_8u\_C3IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical and.*
- **NppStatus** **nppiAnd\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and with unmodified alpha.*
- **NppStatus** **nppiAnd\_8u\_AC4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and with unmodified alpha.*
- **NppStatus** **nppiAnd\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical and.*
- **NppStatus** **nppiAnd\_8u\_C4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical and.*
- **NppStatus** **nppiAnd\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical and.*
- **NppStatus** **nppiAnd\_16u\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical and.*
- **NppStatus** **nppiAnd\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical and.*

- **NppStatus nppiAnd\_16u\_C3IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image logical and.*
- **NppStatus nppiAnd\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical and with unmodified alpha.*
- **NppStatus nppiAnd\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and with unmodified alpha.*
- **NppStatus nppiAnd\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical and.*
- **NppStatus nppiAnd\_16u\_C4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical and.*
- **NppStatus nppiAnd\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel image logical and.*
- **NppStatus nppiAnd\_32s\_C1IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel in place image logical and.*
- **NppStatus nppiAnd\_32s\_C3R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel image logical and.*
- **NppStatus nppiAnd\_32s\_C3IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel in place image logical and.*
- **NppStatus nppiAnd\_32s\_AC4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical and with unmodified alpha.*
- **NppStatus nppiAnd\_32s\_AC4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical and with unmodified alpha.*
- **NppStatus nppiAnd\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical and.*
- **NppStatus nppiAnd\_32s\_C4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical and.*

### 7.34.1 Detailed Description

Pixel by pixel logical and of images.

### 7.34.2 Function Documentation

#### 7.34.2.1 `NppStatus nppiAnd_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical and with unmodified alpha.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.34.2.2 `NppStatus nppiAnd_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical and with unmodified alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.34.2.3 `NppStatus nppiAnd_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical and.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.34.2.4 NppStatus nppiAnd\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.34.2.5 NppStatus nppiAnd\_16u\_C3IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.34.2.6 NppStatus nppiAnd\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image logical and.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.34.2.7 NppStatus nppiAnd\_16u\_C4IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical and.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.34.2.8 NppStatus nppiAnd\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image logical and.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.9 NppStatus nppiAnd\_32s\_AC4IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel in place image logical and with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.10 NppStatus nppiAnd\_32s\_AC4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 32-bit signed integer channel image logical and with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.11 NppStatus nppiAnd\_32s\_C1IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.12 NppStatus nppiAnd\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit signed integer channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.13 NppStatus nppiAnd\_32s\_C3IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.34.2.14 **NppStatus nppiAnd\_32s\_C3R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical and.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.34.2.15 **NppStatus nppiAnd\_32s\_C4IR** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical and.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.34.2.16 **NppStatus nppiAnd\_32s\_C4R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical and.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.17 NppStatus nppiAnd\_8u\_AC4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel in place image logical and with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.18 NppStatus nppiAnd\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel image logical and with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.34.2.19 NppStatus nppiAnd\_8u\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical and.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.34.2.20 NppStatus nppiAnd\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical and.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.34.2.21 NppStatus nppiAnd\_8u\_C3IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical and.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.34.2.22** `NppStatus nppiAnd_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.23** `NppStatus nppiAnd_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical and.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.34.2.24** `NppStatus nppiAnd_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical and.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.35 Or

Pixel by pixel logical or of images.

### Functions

- **NppStatus nppiOr\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical or.*
- **NppStatus nppiOr\_8u\_C1IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical or.*
- **NppStatus nppiOr\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical or.*
- **NppStatus nppiOr\_8u\_C3IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical or.*
- **NppStatus nppiOr\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or with unmodified alpha.*
- **NppStatus nppiOr\_8u\_AC4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or with unmodified alpha.*
- **NppStatus nppiOr\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical or.*
- **NppStatus nppiOr\_8u\_C4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical or.*
- **NppStatus nppiOr\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image logical or.*
- **NppStatus nppiOr\_16u\_C1IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image logical or.*
- **NppStatus nppiOr\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image logical or.*



- `NppStatus nppiOr_16u_C3IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel in place image logical or.*
- `NppStatus nppiOr_16u_AC4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or with unmodified alpha.*
- `NppStatus nppiOr_16u_AC4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or with unmodified alpha.*
- `NppStatus nppiOr_16u_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel image logical or.*
- `NppStatus nppiOr_16u_C4IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 16-bit unsigned short channel in place image logical or.*
- `NppStatus nppiOr_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel image logical or.*
- `NppStatus nppiOr_32s_C1IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 32-bit signed integer channel in place image logical or.*
- `NppStatus nppiOr_32s_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel image logical or.*
- `NppStatus nppiOr_32s_C3IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 32-bit signed integer channel in place image logical or.*
- `NppStatus nppiOr_32s_AC4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or with unmodified alpha.*
- `NppStatus nppiOr_32s_AC4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or with unmodified alpha.*
- `NppStatus nppiOr_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel image logical or.*
- `NppStatus nppiOr_32s_C4IR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 32-bit signed integer channel in place image logical or.*

### 7.35.1 Detailed Description

Pixel by pixel logical or of images.

### 7.35.2 Function Documentation

#### 7.35.2.1 `NppStatus nppiOr_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.35.2.2 `NppStatus nppiOr_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.35.2.3 `NppStatus nppiOr_16u_C1IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.35.2.4 NppStatus nppiOr\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.35.2.5 NppStatus nppiOr\_16u\_C3IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.35.2.6 NppStatus nppiOr\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image logical or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.35.2.7 NppStatus nppiOr\_16u\_C4IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.35.2.8 NppStatus nppiOr\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image logical or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.9 NppStatus nppiOr\_32s\_AC4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.10 NppStatus nppiOr\_32s\_AC4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.11 NppStatus nppiOr\_32s\_C1IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.12 NppStatus nppiOr\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 32-bit signed integer channel image logical or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.35.2.13 NppStatus nppiOr\_32s\_C3IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 32-bit signed integer channel in place image logical or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.35.2.14 `NppStatus nppiOr_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three 32-bit signed integer channel image logical or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.35.2.15 `NppStatus nppiOr_32s_C4IR (const Npp32s * pSrc, int nSrcStep, Npp32s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel in place image logical or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.35.2.16 `NppStatus nppiOr_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four 32-bit signed integer channel image logical or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.17 NppStatus nppiOr\_8u\_AC4IR (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel in place image logical or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.35.2.18 NppStatus nppiOr\_8u\_AC4R (const Npp8u \*pSrc1, int nSrc1Step, const Npp8u \*pSrc2, int nSrc2Step, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image logical or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.35.2.19 NppStatus nppiOr\_8u\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.35.2.20 NppStatus nppiOr\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.35.2.21 NppStatus nppiOr\_8u\_C3IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.35.2.22 NppStatus nppiOr\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image logical or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.35.2.23 NppStatus nppiOr\_8u\_C4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image logical or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.35.2.24 NppStatus nppiOr\_8u\_C4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image logical or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.36 Xor

Pixel by pixel logical exclusive or of images.

### Functions

- `NppStatus nppiXor_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel image logical exclusive or.*
- `NppStatus nppiXor_8u_C1IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 8-bit unsigned char channel in place image logical exclusive or.*
- `NppStatus nppiXor_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel image logical exclusive or.*
- `NppStatus nppiXor_8u_C3IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Three 8-bit unsigned char channel in place image logical exclusive or.*
- `NppStatus nppiXor_8u_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.*
- `NppStatus nppiXor_8u_AC4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.*
- `NppStatus nppiXor_8u_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel image logical exclusive or.*
- `NppStatus nppiXor_8u_C4IR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*Four 8-bit unsigned char channel in place image logical exclusive or.*
- `NppStatus nppiXor_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel image logical exclusive or.*
- `NppStatus nppiXor_16u_C1IR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI)  
*One 16-bit unsigned short channel in place image logical exclusive or.*
- `NppStatus nppiXor_16u_C3R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three 16-bit unsigned short channel image logical exclusive or.*

- **NppStatus nppiXor\_16u\_C3IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel in place image logical exclusive or.*
- **NppStatus nppiXor\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel image logical exclusive or.*
- **NppStatus nppiXor\_16u\_C4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 16-bit unsigned short channel in place image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C1IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 32-bit signed integer channel in place image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C3R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C3IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 32-bit signed integer channel in place image logical exclusive or.*
- **NppStatus nppiXor\_32s\_AC4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_32s\_AC4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.*
- **NppStatus nppiXor\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel image logical exclusive or.*
- **NppStatus nppiXor\_32s\_C4IR** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 32-bit signed integer channel in place image logical exclusive or.*

### 7.36.1 Detailed Description

Pixel by pixel logical exclusive or of images.

### 7.36.2 Function Documentation

#### 7.36.2.1 `NppStatus nppiXor_16u_AC4IR (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel in place image logical exclusive or with unmodified alpha.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.36.2.2 `NppStatus nppiXor_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 16-bit unsigned short channel image logical exclusive or with unmodified alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.36.2.3 `NppStatus nppiXor_16u_C11R (const Npp16u * pSrc, int nSrcStep, Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One 16-bit unsigned short channel in place image logical exclusive or.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.36.2.4 NppStatus nppiXor\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 16-bit unsigned short channel image logical exclusive or.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.36.2.5 NppStatus nppiXor\_16u\_C3IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image logical exclusive or.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.36.2.6 NppStatus nppiXor\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.7 NppStatus nppiXor\_16u\_C4IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image logical exclusive or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.8 NppStatus nppiXor\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.9 NppStatus nppiXor\_32s\_AC4IR (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.10 NppStatus nppiXor\_32s\_AC4R (const Npp32s \* pSrc1, int nSrc1Step, const Npp32s \* pSrc2, int nSrc2Step, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 32-bit signed integer channel image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.36.2.11 **NppStatus nppiXor\_32s\_C1IR** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel in place image logical exclusive or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.12 **NppStatus nppiXor\_32s\_C1R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 32-bit signed integer channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.13 **NppStatus nppiXor\_32s\_C3IR** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel in place image logical exclusive or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.14 NppStatus nppiXor\_32s\_C3R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 32-bit signed integer channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.15 NppStatus nppiXor\_32s\_C4IR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel in place image logical exclusive or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.16 NppStatus nppiXor\_32s\_C4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 32-bit signed integer channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.17 NppStatus nppiXor\_8u\_AC4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel in place image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.36.2.18 NppStatus nppiXor\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel image logical exclusive or with unmodified alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.36.2.19 NppStatus nppiXor\_8u\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image logical exclusive or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.20 NppStatus nppiXor\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.21 NppStatus nppiXor\_8u\_C3IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image logical exclusive or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.22 `NppStatus nppiXor_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three 8-bit unsigned char channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.23 `NppStatus nppiXor_8u_C4IR (const Npp8u * pSrc, int nSrcStep, Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel in place image logical exclusive or.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.36.2.24 `NppStatus nppiXor_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four 8-bit unsigned char channel image logical exclusive or.

#### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.37 Not

Pixel by pixel logical not of image.

### Functions

- **NppStatus nppiNot\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image logical not.*
- **NppStatus nppiNot\_8u\_C1IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image logical not.*
- **NppStatus nppiNot\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image logical not.*
- **NppStatus nppiNot\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image logical not.*
- **NppStatus nppiNot\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical not with unmodified alpha.*
- **NppStatus nppiNot\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical not with unmodified alpha.*
- **NppStatus nppiNot\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image logical not.*
- **NppStatus nppiNot\_8u\_C4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image logical not.*

### 7.37.1 Detailed Description

Pixel by pixel logical not of image.

### 7.37.2 Function Documentation

#### 7.37.2.1 NppStatus nppiNot\_8u\_AC4IR (Npp8u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)

Four 8-bit unsigned char channel in place image logical not with unmodified alpha.

#### Parameters:

*pSrcDst* **In-Place Image Pointer.**

*nSrcDstStep* **In-Place-Image Line Step.**



*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.2 NppStatus nppiNot\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four 8-bit unsigned char channel image logical not with unmodified alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.3 NppStatus nppiNot\_8u\_C1IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.4 NppStatus nppiNot\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

One 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.37.2.5 NppStatus nppiNot\_8u\_C3IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.37.2.6 NppStatus nppiNot\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.37.2.7 NppStatus nppiNot\_8u\_C4IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel in place image logical not.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.37.2.8 NppStatus nppiNot\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four 8-bit unsigned char channel image logical not.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.38 Alpha Composition

### Modules

- [AlphaCompC](#)  
*Composite two images using constant alpha values.*
- [AlphaPremulC](#)  
*Premultiplies pixels of an image using a constant alpha value.*
- [AlphaComp](#)  
*Composite two images using alpha opacity values contained in each image.*
- [AlphaPremul](#)  
*Premultiplies image pixels by image alpha opacity values.*

## 7.39 AlphaCompC

Composite two images using constant alpha values.

### Functions

- **NppStatus nppiAlphaCompC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*One 8-bit unsigned char channel image composition using constant alpha.*

- **NppStatus nppiAlphaCompC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*Three 8-bit unsigned char channel image composition using constant alpha.*

- **NppStatus nppiAlphaCompC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*Four 8-bit unsigned char channel image composition using constant alpha.*

- **NppStatus nppiAlphaCompC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*Four 8-bit unsigned char channel image composition with alpha using constant source alpha.*

- **NppStatus nppiAlphaCompC\_8s\_C1R** (const **Npp8s** \*pSrc1, int nSrc1Step, **Npp8s** nAlpha1, const **Npp8s** \*pSrc2, int nSrc2Step, **Npp8s** nAlpha2, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*One 8-bit signed char channel image composition using constant alpha.*

- **NppStatus nppiAlphaCompC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** nAlpha2, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*One 16-bit unsigned short channel image composition using constant alpha.*

- **NppStatus nppiAlphaCompC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** nAlpha2, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*Three 16-bit unsigned short channel image composition using constant alpha.*

- **NppStatus nppiAlphaCompC\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** nAlpha2, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*Four 16-bit unsigned short channel image composition using constant alpha.*

- **NppStatus nppiAlphaCompC\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** nAlpha2, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)

*Four 16-bit unsigned short channel image composition with alpha using constant source alpha.*

- `NppStatus nppiAlphaCompC_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, `Npp16s` nAlpha1, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` nAlpha2, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

*One 16-bit signed short channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_32u_C1R` (const `Npp32u` \*pSrc1, int nSrc1Step, `Npp32u` nAlpha1, const `Npp32u` \*pSrc2, int nSrc2Step, `Npp32u` nAlpha2, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

*One 32-bit unsigned integer channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, `Npp32s` nAlpha1, const `Npp32s` \*pSrc2, int nSrc2Step, `Npp32s` nAlpha2, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

*One 32-bit signed integer channel image composition using constant alpha.*

- `NppStatus nppiAlphaCompC_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, `Npp32f` nAlpha1, const `Npp32f` \*pSrc2, int nSrc2Step, `Npp32f` nAlpha2, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

*One 32-bit floating point channel image composition using constant alpha.*

### 7.39.1 Detailed Description

Composite two images using constant alpha values.

### 7.39.2 Function Documentation

#### 7.39.2.1 `NppStatus nppiAlphaCompC_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, `Npp16s` nAlpha1, const `Npp16s` \*pSrc2, int nSrc2Step, `Npp16s` nAlpha2, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppiAlphaOp` eAlphaOp)

One 16-bit signed short channel image composition using constant alpha.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*nAlpha2* Image alpha opacity (0 - max channel pixel value).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eAlphaOp* alpha-blending operation..

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.39.2.2 NppStatus nppiAlphaCompC\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 16-bit unsigned short channel image composition with alpha using constant source alpha.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.39.2.3 NppStatus nppiAlphaCompC\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 16-bit unsigned short channel image composition using constant alpha.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.39.2.4 NppStatus nppiAlphaCompC\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Three 16-bit unsigned short channel image composition using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.39.2.5 NppStatus nppiAlphaCompC\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp16u *nAlpha2*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 16-bit unsigned short channel image composition using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.39.2.6 NppStatus nppiAlphaCompC\_32f\_C1R (const Npp32f \* *pSrc1*, int *nSrc1Step*, Npp32f *nAlpha1*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f *nAlpha2*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)**

One 32-bit floating point channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0.0 - 1.0).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0.0 - 1.0).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.7 NppStatus nppiAlphaCompC\_32s\_C1R (const Npp32s \* *pSrc1*, int *nSrc1Step*, Npp32s *nAlpha1*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s *nAlpha2*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)**

One 32-bit signed integer channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.39.2.8 NppStatus nppiAlphaCompC\_32u\_C1R (const Npp32u \* *pSrc1*, int *nSrc1Step*, Npp32u *nAlpha1*, const Npp32u \* *pSrc2*, int *nSrc2Step*, Npp32u *nAlpha2*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit unsigned integer channel image composition using constant alpha.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.39.2.9 NppStatus nppiAlphaCompC\_8s\_C1R (const Npp8s \* *pSrc1*, int *nSrc1Step*, Npp8s *nAlpha1*, const Npp8s \* *pSrc2*, int *nSrc2Step*, Npp8s *nAlpha2*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit signed char channel image composition using constant alpha.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.10** `NppStatus nppiAlphaCompC_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition with alpha using constant source alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.11** `NppStatus nppiAlphaCompC_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

One 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.12** `NppStatus nppiAlphaCompC_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Three 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.39.2.13** `NppStatus nppiAlphaCompC_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, Npp8u nAlpha1, const Npp8u * pSrc2, int nSrc2Step, Npp8u nAlpha2, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition using constant alpha.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*nAlpha2* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* alpha-blending operation..

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.40 AlphaPremulC

Premultiplies pixels of an image using a constant alpha value.

### Functions

- **NppStatus nppiAlphaPremulC\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_8u\_C1IR** (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 8-bit unsigned char channel in place image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_8u\_C3IR** (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Three 8-bit unsigned char channel in place image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_8u\_C4IR** (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.*
- **NppStatus nppiAlphaPremulC\_8u\_AC4IR** (**Npp8u** nAlpha1, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.*
- **NppStatus nppiAlphaPremulC\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_16u\_C1IR** (**Npp16u** nAlpha1, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*One 16-bit unsigned short channel in place image premultiplication using constant alpha.*
- **NppStatus nppiAlphaPremulC\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** nAlpha1, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three 16-bit unsigned short channel image premultiplication using constant alpha.*

- `NppStatus nppiAlphaPremulC_16u_C3IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)  
*Three 16-bit unsigned short channel in place image premultiplication using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_C4R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `Npp16u nAlpha1`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel image premultiplication using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_C4IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel in place image premultiplication using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_AC4R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `Npp16u nAlpha1`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.*
- `NppStatus nppiAlphaPremulC_16u_AC4IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)  
*Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.*

### 7.40.1 Detailed Description

Premultiplies pixels of an image using a constant alpha value.

### 7.40.2 Function Documentation

#### 7.40.2.1 `NppStatus nppiAlphaPremulC_16u_AC4IR` (`Npp16u nAlpha1`, `Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`)

Four 16-bit unsigned short channel in place image premultiplication with alpha using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.40.2.2 `NppStatus nppiAlphaPremulC_16u_AC4R` (`const Npp16u *pSrc1`, `int nSrc1Step`, `Npp16u nAlpha1`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`)

Four 16-bit unsigned short channel image premultiplication with alpha using constant alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.40.2.3 NppStatus nppiAlphaPremulC\_16u\_C1IR (Npp16u *nAlpha1*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel in place image premultiplication using constant alpha.

**Parameters:**

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.40.2.4 NppStatus nppiAlphaPremulC\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

One 16-bit unsigned short channel image premultiplication using constant alpha.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*nAlpha1* Image alpha opacity (0 - max channel pixel value).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.40.2.5 **NppStatus nppiAlphaPremulC\_16u\_C3IR** (Npp16u *nAlpha1*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.6 **NppStatus nppiAlphaPremulC\_16u\_C3R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 16-bit unsigned short channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.7 **NppStatus nppiAlphaPremulC\_16u\_C4IR** (Npp16u *nAlpha1*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



#### 7.40.2.8 NppStatus nppiAlphaPremulC\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u *nAlpha1*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.9 NppStatus nppiAlphaPremulC\_8u\_AC4IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image premultiplication with alpha using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.10 NppStatus nppiAlphaPremulC\_8u\_AC4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication with alpha using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.11 NppStatus nppiAlphaPremulC\_8u\_C1IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.12 NppStatus nppiAlphaPremulC\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

One 8-bit unsigned char channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.13 NppStatus nppiAlphaPremulC\_8u\_C3IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.14 NppStatus nppiAlphaPremulC\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three 8-bit unsigned char channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.15 NppStatus nppiAlphaPremulC\_8u\_C4IR (Npp8u *nAlpha1*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image premultiplication using constant alpha.

##### Parameters:

*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pSrcDst* [In-Place Image Pointer](#).  
*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.40.2.16 NppStatus nppiAlphaPremulC\_8u\_C4R (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u *nAlpha1*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication using constant alpha.

##### Parameters:

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*nAlpha1* Image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.41 AlphaComp

Composite two images using alpha opacity values contained in each image.

### Functions

- **NppStatus** **nppiAlphaComp\_8u\_AC1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_8s\_AC1R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_16u\_AC1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_16s\_AC1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_32u\_AC1R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **Npp32u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_32u\_AC4R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **Npp32u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus** **nppiAlphaComp\_32s\_AC1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiAlphaOp** eAlphaOp)  
*One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).*

- **NppStatus nppiAlphaComp\_32s\_AC4R** (const [Npp32s](#) \*pSrc1, int nSrc1Step, const [Npp32s](#) \*pSrc2, int nSrc2Step, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)  
*Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).*
- **NppStatus nppiAlphaComp\_32f\_AC1R** (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)  
*One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).*
- **NppStatus nppiAlphaComp\_32f\_AC4R** (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)  
*Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).*

### 7.41.1 Detailed Description

Composite two images using alpha opacity values contained in each image.

### 7.41.2 Function Documentation

#### 7.41.2.1 **NppStatus nppiAlphaComp\_16s\_AC1R** (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)

One 16-bit signed short channel image composition using image alpha values (0 - max channel pixel value).

##### Parameters:

[pSrc1](#) Source-Image Pointer.  
[nSrc1Step](#) Source-Image Line Step.  
[pSrc2](#) Source-Image Pointer.  
[nSrc2Step](#) Source-Image Line Step.  
[pDst](#) Destination-Image Pointer.  
[nDstStep](#) Destination-Image Line Step.  
[oSizeROI](#) Region-of-Interest (ROI).  
[eAlphaOp](#) alpha-blending operation..

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.41.2.2 **NppStatus nppiAlphaComp\_16u\_AC1R** (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [NppiAlphaOp](#) eAlphaOp)

One 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.41.2.3** `NppStatus nppiAlphaComp_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 16-bit unsigned short channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.41.2.4** `NppStatus nppiAlphaComp_32f_AC1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

One 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.41.2.5 **NppStatus nppiAlphaComp\_32f\_AC4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 32-bit floating point channel image composition using image alpha values (0.0 - 1.0).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.41.2.6 **NppStatus nppiAlphaComp\_32s\_AC1R** (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.41.2.7 NppStatus nppiAlphaComp\_32s\_AC4R (const Npp32s \* *pSrc1*, int *nSrc1Step*, const Npp32s \* *pSrc2*, int *nSrc2Step*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)**

Four 32-bit signed integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.41.2.8 NppStatus nppiAlphaComp\_32u\_AC1R (const Npp32u \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pSrc2*, int *nSrc2Step*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)**

One 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.41.2.9 NppStatus nppiAlphaComp\_32u\_AC4R** (const Npp32u \* *pSrc1*, int *nSrc1Step*, const Npp32u \* *pSrc2*, int *nSrc2Step*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

Four 32-bit unsigned integer channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* [alpha-blending operation..](#)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.10 NppStatus nppiAlphaComp\_8s\_AC1R** (const Npp8s \* *pSrc1*, int *nSrc1Step*, const Npp8s \* *pSrc2*, int *nSrc2Step*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit signed char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eAlphaOp* [alpha-blending operation..](#)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.41.2.11 NppStatus nppiAlphaComp\_8u\_AC1R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppiAlphaOp *eAlphaOp*)

One 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.41.2.12** `NppStatus nppiAlphaComp_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppiAlphaOp eAlphaOp)`

Four 8-bit unsigned char channel image composition using image alpha values (0 - max channel pixel value).

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eAlphaOp* alpha-blending operation..

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.42 AlphaPremul

Premultiplies image pixels by image alpha opacity values.

### Functions

- **NppStatus nppiAlphaPremul\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).*

- **NppStatus nppiAlphaPremul\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).*

- **NppStatus nppiAlphaPremul\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).*

- **NppStatus nppiAlphaPremul\_16u\_AC4IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).*

### 7.42.1 Detailed Description

Premultiplies image pixels by image alpha opacity values.

### 7.42.2 Function Documentation

#### 7.42.2.1 **NppStatus nppiAlphaPremul\_16u\_AC4IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four 16-bit unsigned short channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

#### Parameters:

**pSrcDst** [In-Place Image Pointer](#).

**nSrcDstStep** [In-Place-Image Line Step](#).

**oSizeROI** [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.42.2.2 **NppStatus nppiAlphaPremul\_16u\_AC4R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 16-bit unsigned short channel image premultiplication with pixel alpha (0 - max channel pixel value).

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.42.2.3 **NppStatus nppiAlphaPremul\_8u\_AC4IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel in place image premultiplication with pixel alpha (0 - max channel pixel value).

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.42.2.4 **NppStatus nppiAlphaPremul\_8u\_AC4R** (const Npp8u \* *pSrc1*, int *nSrc1Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four 8-bit unsigned char channel image premultiplication with pixel alpha (0 - max channel pixel value).

##### Parameters:

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

## 7.43 Color and Sampling Conversion

Routines manipulating an image's color model and sampling format.

### Modules

- [Color Model Conversion](#)

*Routines for converting between various image color models.*

- [Color Sampling Format Conversion](#)

*Routines for converting between various image color sampling formats.*

- [Color Gamma Correction](#)

*Routines for correcting image color gamma.*

- [Complement Color Key](#)

*Routines for performing complement color key replacement.*

- [Color Processing](#)

*Routines for performing image color manipulation.*

### 7.43.1 Detailed Description

Routines manipulating an image's color model and sampling format.

## 7.44 Color Model Conversion

Routines for converting between various image color models.

### RGBToYUV

RGB to YUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YUV. For digital RGB values in the range [0..255], Y has the range [0..255], U varies in the range [-112..+112], and V in the range [-157..+157]. To fit in the range of [0..255], a constant value of 128 is added to computed U and V values, and V is then saturated.

```
Npp32f nY = 0.299F * R + 0.587F * G + 0.114F * B;
Npp32f nU = (0.492F * ((Npp32f)nB - nY)) + 128.0F;
Npp32f nV = (0.877F * ((Npp32f)nR - nY)) + 128.0F;
if (nV > 255.0F)
    nV = 255.0F;
```

- **NppStatus nppiRGBToYUV\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YUV color conversion.*

- **NppStatus nppiRGBToYUV\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.*

- **NppStatus nppiRGBToYUV\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV color conversion.*

- **NppStatus nppiRGBToYUV\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV color conversion.*

- **NppStatus nppiRGBToYUV\_8u\_AC4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.*

### BGRToYUV

BGR to YUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YUV. For digital RGB values in the range [0..255], Y has the range [0..255], U varies in the range [-112..+112], and V in the range [-157..+157]. To fit in the range of [0..255], a constant value of 128 is added to computed U and V values, and V is then saturated.

```

Npp32f nY = 0.299F * R + 0.587F * G + 0.114F * B;
Npp32f nU = (0.492F * ((Npp32f)nB - nY)) + 128.0F;
Npp32f nV = (0.877F * ((Npp32f)nR - nY)) + 128.0F;
if (nV > 255.0F)
    nV = 255.0F;

```

- **NppStatus nppiBGRToYUV\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed YUV color conversion.*

- **NppStatus nppiBGRToYUV\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.*

- **NppStatus nppiBGRToYUV\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YUV color conversion.*

- **NppStatus nppiBGRToYUV\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YUV color conversion.*

- **NppStatus nppiBGRToYUV\_8u\_AC4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.*

## YUVToRGB

YUV to RGB color conversion.

Here is how NPP converts YUV to gamma corrected RGB or BGR.

```

Npp32f nY = (Npp32f)Y;
Npp32f nU = (Npp32f)U - 128.0F;
Npp32f nV = (Npp32f)V - 128.0F;
Npp32f nR = nY + 1.140F * nV;
if (nR < 0.0F)
    nR = 0.0F;
if (nR > 255.0F)
    nR = 255.0F;
Npp32f nG = nY - 0.394F * nU - 0.581F * nV;
if (nG < 0.0F)
    nG = 0.0F;
if (nG > 255.0F)
    nG = 255.0F;
Npp32f nB = nY + 2.032F * nU;
if (nB < 0.0F)
    nB = 0.0F;
if (nB > 255.0F)
    nB = 255.0F;

```

- **NppStatus nppiYUVToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus nppiYUVToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed RGB color conversion with alpha, not affecting alpha.*
- **NppStatus nppiYUVToRGB\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar RGB color conversion.*
- **NppStatus nppiYUVToRGB\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB color conversion.*

## YUVToBGR

YUV to BGR color conversion.

Here is how NPP converts YUV to gamma corrected RGB or BGR.

```
Npp32f nY = (Npp32f) Y;
Npp32f nU = (Npp32f) U - 128.0F;
Npp32f nV = (Npp32f) V - 128.0F;
Npp32f nR = nY + 1.140F * nV;
if (nR < 0.0F)
    nR = 0.0F;
if (nR > 255.0F)
    nR = 255.0F;
Npp32f nG = nY - 0.394F * nU - 0.581F * nV;
if (nG < 0.0F)
    nG = 0.0F;
if (nG > 255.0F)
    nG = 255.0F;
Npp32f nB = nY + 2.032F * nU;
if (nB < 0.0F)
    nB = 0.0F;
if (nB > 255.0F)
    nB = 255.0F;
```

- **NppStatus nppiYUVToBGR\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed BGR color conversion.*
- **NppStatus nppiYUVToBGR\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed BGR color conversion with alpha, not affecting alpha.*
- **NppStatus nppiYUVToBGR\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar BGR color conversion.*
- **NppStatus nppiYUVToBGR\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)



*3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed BGR color conversion.*

## RGBToYUV422

RGB to YUV422 color conversion.

- `NppStatus nppiRGBToYUV422_8u_C3C2R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YUV422 color conversion.*
- `NppStatus nppiRGBToYUV422_8u_P3R` (const `Npp8u` \*const pSrc[3], int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.*
- `NppStatus nppiRGBToYUV422_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.*

## YUV422ToRGB

YUV422 to RGB color conversion.

- `NppStatus nppiYUV422ToRGB_8u_C2C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned packed YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.*
- `NppStatus nppiYUV422ToRGB_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned planar RGB color conversion.*
- `NppStatus nppiYUV422ToRGB_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.*
- `NppStatus nppiYUV422ToRGB_8u_P3AC4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YUV422 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.*

## RGBToYUV420

RGB to YUV420 color conversion.

- `NppStatus nppiRGBToYUV420_8u_P3R` (const `Npp8u` \*const pSrc[3], int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.*

- `NppStatus nppiRGBToYUV420_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.*

## YUV420ToRGB

YUV420 to RGB color conversion.

- `NppStatus nppiYUV420ToRGB_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned planar RGB color conversion.*

- `NppStatus nppiYUV420ToRGB_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB color conversion.*

- `NppStatus nppiYUV420ToRGB_8u_P3AC4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.*

## BGRToYUV420

BGR to YUV420 color conversion.

- `NppStatus nppiBGRToYUV420_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YUV420 color conversion.*

## YUV420ToBGR

YUV420 to BGR color conversion.

- `NppStatus nppiYUV420ToBGR_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR color conversion.*

## RGBToYCbCr

RGB to YCbCr color conversion.

Here is how NPP converts gamma corrected RGB or BGR to YCbCr. In the YCbCr model, Y is defined to have a nominal range [16..235], while Cb and Cr are defined to have a range [16..240], with the value of 128 as corresponding to zero.

```

Npp32f nY  = 0.257F * R + 0.504F * G + 0.098F * B + 16.0F;
Npp32f nCb = -0.148F * R - 0.291F * G + 0.439F * B + 128.0F;
Npp32f nCr  = 0.439F * R - 0.368F * G - 0.071F * B + 128.0F;

```

- **NppStatus nppiRGBToYCbCr\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit packed YCbCr color conversion.*
- **NppStatus nppiRGBToYCbCr\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel unsigned 8-bit packed YCbCr with alpha color conversion, not affecting alpha.*
- **NppStatus nppiRGBToYCbCr\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel planar 8-bit unsigned RGB to 3 channel planar 8-bit YCbCr color conversion.*
- **NppStatus nppiRGBToYCbCr\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit planar YCbCr color conversion.*
- **NppStatus nppiRGBToYCbCr\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.*

## YCbCrToRGB

YCbCr to RGB color conversion.

Here is how NPP converts YCbCr to gamma corrected RGB or BGR. The output RGB values are saturated to the range [0..255].

```

Npp32f nY = 1.164F * ((Npp32f)Y - 16.0F);
Npp32f nR = ((Npp32f)Cr - 128.0F);
Npp32f nB = ((Npp32f)Cb - 128.0F);
Npp32f nG = nY - 0.813F * nR - 0.392F * nB;
if (nG > 255.0F)
    nG = 255.0F;
nR = nY + 1.596F * nR;
if (nR > 255.0F)
    nR = 255.0F;
nB = nY + 2.017F * nB;
if (nB > 255.0F)
    nB = 255.0F;

```

- **NppStatus nppiYCbCrToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiYCbCrToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed YCbCr with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion, not affecting alpha.*

- **NppStatus nppiYCbCrToRGB\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned planar RGB color conversion.*
- **NppStatus nppiYCbCrToRGB\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiYCbCrToRGB\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.*

## YCbCrToBGR

YCbCr to BGR color conversion.

- **NppStatus nppiYCbCrToBGR\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR color conversion.*
- **NppStatus nppiYCbCrToBGR\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*

## YCbCrToBGR\_709CSC

YCbCr to BGR\_709CSC color conversion.

- **NppStatus nppiYCbCrToBGR\_709CSC\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.*
- **NppStatus nppiYCbCrToBGR\_709CSC\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR\_709CSC color conversion with constant alpha.*

## RGBToYCbCr422

RGB to YCbCr422 color conversion.

- **NppStatus nppiRGBToYCbCr422\_8u\_C3C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.*

- **NppStatus** **nppiRGBToYCbCr422\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion.*

- **NppStatus** **nppiRGBToYCbCr422\_8u\_P3C2R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion.*

## YCbCr422ToRGB

YCbCr422 to RGB color conversion.

- **NppStatus** **nppiYCbCr422ToRGB\_8u\_C2C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus** **nppiYCbCr422ToRGB\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion.*

- **NppStatus** **nppiYCbCr422ToRGB\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion.*

## RGBToYCrCb422

RGB to YCrCb422 color conversion.

- **NppStatus** **nppiRGBToYCrCb422\_8u\_C3C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*

- **NppStatus** **nppiRGBToYCrCb422\_8u\_P3C2R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*

## YCrCb422ToRGB

YCrCb422 to RGB color conversion.

- **NppStatus** **nppiYCrCb422ToRGB\_8u\_C2C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed RGB color conversion.*

- `NppStatus nppiYCrCb422ToRGB_8u_C2P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar RGB color conversion.*

## BGRToYCbCr422

BGR to YCbCr422 color conversion.

- `NppStatus nppiBGRToYCbCr422_8u_C3C2R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*

- `NppStatus nppiBGRToYCbCr422_8u_AC4C2R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed YCrCb422 color conversion.*

- `NppStatus nppiBGRToYCbCr422_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int nDstStep[3], `NppiSize` oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion.*

- `NppStatus nppiBGRToYCbCr422_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int nDstStep[3], `NppiSize` oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr422 color conversion.*

## YCbCr422ToBGR

YCbCr422 to BGR color conversion.

- `NppStatus nppiYCbCr422ToBGR_8u_C2C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed BGR color conversion.*

- `NppStatus nppiYCbCr422ToBGR_8u_C2C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)

*2 channel 8-bit unsigned packed YCrCb422 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*

- `NppStatus nppiYCbCr422ToBGR_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion.*

## RGBToCbYCr422

RGB to CbYCr422 color conversion.

- **NppStatus nppiRGBToCbYCr422\_8u\_C3C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed CbYCr422 color conversion.*
- **NppStatus nppiRGBToCbYCr422Gamma\_8u\_C3C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB first gets forward gamma corrected then converted to 2 channel 8-bit unsigned packed CbYCr422 color conversion.*

## CbYCr422ToRGB

CbYCr422 to RGB color conversion.

- **NppStatus nppiCbYCr422ToRGB\_8u\_C2C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed RGB color conversion.*

## BGRToCbYCr422

BGR to CbYCr422 color conversion.

- **NppStatus nppiBGRToCbYCr422\_8u\_AC4C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422 color conversion.*

## BGRToCbYCr422\_709HDTV

BGR to CbYCr422\_709HDTV color conversion.

- **NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_C3C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.*
- **NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_AC4C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.*

## CbYCr422ToBGR

CbYCr422 to BGR color conversion.

- **NppStatus nppiCbYCr422ToBGR\_8u\_C2C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)

*2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR color conversion with alpha.*

## CbYCr422ToBGR\_709HDTV

CbYCr422 to BGR\_709HDTV color conversion.

- [NppStatus nppiCbYCr422ToBGR\\_709HDTV\\_8u\\_C2C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed BGR\_709HDTV color conversion.*

- [NppStatus nppiCbYCr422ToBGR\\_709HDTV\\_8u\\_C2C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, [Npp8u](#) nAval)

*2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.*

## RGBToYCbCr420

RGB to YCbCr420 color conversion.

- [NppStatus nppiRGBToYCbCr420\\_8u\\_C3P3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst[3], int rDstStep[3], [NppiSize](#) oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion.*

## YCbCr420ToRGB

YCbCr420 to RGB color conversion.

- [NppStatus nppiYCbCr420ToRGB\\_8u\\_P3C3R](#) (const [Npp8u](#) \*const pSrc[3], int rSrcStep[3], [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB color conversion.*

## RGBToYCrCb420

RGB to YCrCb420 color conversion.

- [NppStatus nppiRGBToYCrCb420\\_8u\\_AC4P3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst[3], int rDstStep[3], [NppiSize](#) oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.*



## YCrCb420ToRGB

YCrCb420 to RGB color conversion.

- **NppStatus nppiYCrCb420ToRGB\_8u\_P3C4R** (const **Npp8u** \*pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)

*3 channel 8-bit unsigned planar YCrCb420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.*

## BGRToYCbCr420

BGR to YCbCr420 color conversion.

- **NppStatus nppiBGRToYCbCr420\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion.*

- **NppStatus nppiBGRToYCbCr420\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420 color conversion.*

## BGRToYCbCr420\_709CSC

BGR to YCbCr420\_709CSC color conversion.

- **NppStatus nppiBGRToYCbCr420\_709CSC\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion.*

- **NppStatus nppiBGRToYCbCr420\_709CSC\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion.*

## BGRToYCbCr420\_709HDTV

BGR to YCbCr420\_709HDTV color conversion.

- **NppStatus nppiBGRToYCbCr420\_709HDTV\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709HDTV color conversion.*

## BGRToYCrCb420\_709CSC

BGR to YCrCb420\_709CSC color conversion.

- **NppStatus** **nppiBGRToYCrCb420\_709CSC\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.*
- **NppStatus** **nppiBGRToYCrCb420\_709CSC\_8u\_AC4P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.*

## YCbCr420ToBGR

YCbCr420 to BGR color conversion.

- **NppStatus** **nppiYCbCr420ToBGR\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.*
- **NppStatus** **nppiYCbCr420ToBGR\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*

## YCbCr420ToBGR\_709CSC

YCbCr420\_709CSC to BGR color conversion.

- **NppStatus** **nppiYCbCr420ToBGR\_709CSC\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.*

## YCbCr420ToBGR\_709HDTV

YCbCr420\_709HDTV to BGR color conversion.

- **NppStatus** **nppiYCbCr420ToBGR\_709HDTV\_8u\_P3C4R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nAval)  
*3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.*

## BGRToYCrCb420

BGR to YCrCb420 color conversion.

- `NppStatus nppiBGRToYCrCb420_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420 color conversion.*
- `NppStatus nppiBGRToYCrCb420_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.*

## BGRToYCbCr411

BGR to YCbCr411 color conversion.

- `NppStatus nppiBGRToYCbCr411_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion.*
- `NppStatus nppiBGRToYCbCr411_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr411 color conversion.*

## BGRToYCbCr

BGR to YCbCr color conversion.

- `NppStatus nppiBGRToYCbCr_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr color conversion.*
- `NppStatus nppiBGRToYCbCr_8u_AC4P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.*
- `NppStatus nppiBGRToYCbCr_8u_AC4P4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst[4], int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YCbCr color conversion.*

## YCbCr411ToBGR

YCbCr411 to BGR color conversion.

- `NppStatus nppiYCbCr411ToBGR_8u_P3C3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.*
- `NppStatus nppiYCbCr411ToBGR_8u_P3C4R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp8u` nAval)  
*3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.*

## RGBToXYZ

RGB to XYZ color conversion.

Here is how NPP converts gamma corrected RGB or BGR to XYZ.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
if (nX > 1.0F)
    nX = 1.0F;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
if (nY > 1.0F)
    nY = 1.0F;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
if (nZ > 1.0F)
    nZ = 1.0F;
X = (Npp8u) (nX * 255.0F);
Y = (Npp8u) (nY * 255.0F);
Z = (Npp8u) (nZ * 255.0F);
```

- `NppStatus nppiRGBToXYZ_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed XYZ color conversion.*
- `NppStatus nppiRGBToXYZ_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed XYZ with alpha color conversion.*

## XYZToRGB

XYZ to RGB color conversion.

Here is how NPP converts XYZ to gamma corrected RGB or BGR. The code assumes that X,Y, and Z values are in the range [0..1].

```
Npp32f nNormalizedX = (Npp32f)X * 0.003921569F; // / 255.0F
Npp32f nNormalizedY = (Npp32f)Y * 0.003921569F;
Npp32f nNormalizedZ = (Npp32f)Z * 0.003921569F;
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
```

```

if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);
B = (Npp8u) (nB * 255.0F);

```

- **NppStatus nppiXYZToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed XYZ to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus nppiXYZToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed XYZ with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## RGBToLUV

RGB to LUV color conversion.

Here is how NPP converts gamma corrected RGB or BGR to CIE LUV using the CIE XYZ D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], U component in the range [-134..220], and V component in the range [-140..122]. The code uses `cbrtf()` the 32 bit floating point cube root math function.

```

// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvN (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
// Now calculate LUV from the XYZ value
Npp32f nTemp = nX + 15.0F * nY + 3.0F * nZ;
Npp32f nu = 4.0F * nX / nTemp;
Npp32f nv = 9.0F * nY / nTemp;
Npp32f nL = 116.0F * cbrtf(nY) - 16.0F;
if (nL < 0.0F)
    nL = 0.0F;
if (nL > 100.0F)
    nL = 100.0F;
nTemp = 13.0F * nL;
Npp32f nU = nTemp * (nu - nun);
if (nU < -134.0F)
    nU = -134.0F;
if (nU > 220.0F)
    nU = 220.0F;
Npp32f nV = nTemp * (nv - nvN);
if (nV < -140.0F)
    nV = -140.0F;
if (nV > 122.0F)

```

```

    nV = 122.0F;
    L = (Npp8u) (nL * 255.0F * 0.01F); // / 100.0F
    U = (Npp8u) ((nU + 134.0F) * 255.0F * 0.0028249F); // / 354.0F
    V = (Npp8u) ((nV + 140.0F) * 255.0F * 0.0038168F); // / 262.0F

```

- **NppStatus nppiRGBToLUV\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed LUV color conversion.*

- **NppStatus nppiRGBToLUV\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed LUV with alpha color conversion.*

## LUVToRGB

LUV to RGB color conversion.

Here is how NPP converts CIE LUV to gamma corrected RGB or BGR using the CIE XYZ D65 white point with a Y luminance of 1.0. The code uses powf() the 32 bit floating point power math function.

```

// use CIE D65 chromaticity coordinates
#define nCIE_XYZ_D65_xn 0.312713F
#define nCIE_XYZ_D65_yn 0.329016F
#define nn_DIVISOR (-2.0F * nCIE_XYZ_D65_xn + 12.0F * nCIE_XYZ_D65_yn + 3.0F)
#define nun (4.0F * nCIE_XYZ_D65_xn / nn_DIVISOR)
#define nvN (9.0F * nCIE_XYZ_D65_yn / nn_DIVISOR)

// First convert normalized LUV back to original CIE LUV range
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nU = ((Npp32f)U * 354.0F * 0.003921569F) - 134.0F;
Npp32f nV = ((Npp32f)V * 262.0F * 0.003921569F) - 140.0F;
// Now convert LUV to CIE XYZ
Npp32f nTemp = 13.0F * nL;
Npp32f nu = nU / nTemp + nun;
Npp32f nv = nV / nTemp + nvN;
Npp32f nNormalizedY;
if (nL > 7.9996248F)
{
    nNormalizedY = (nL + 16.0F) * 0.008621F; // / 116.0F
    nNormalizedY = powf(nNormalizedY, 3.0F);
}
else
{
    nNormalizedY = nL * 0.001107F; // / 903.3F
}
Npp32f nNormalizedX = (-9.0F * nNormalizedY * nu) / ((nu - 4.0F) * nv - nu * nv);
Npp32f nNormalizedZ = (9.0F * nNormalizedY - 15.0F * nv * nNormalizedY - nv * nNormalizedX) / (3.0F * nv * nNormalizedY + 15.0F * nv * nNormalizedX - 15.0F * nNormalizedX);
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
if (nR < 0.0F)
    nR = 0.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
if (nG < 0.0F)
    nG = 0.0F;
Npp32f nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;

```

```

if (nB < 0.0F)
    nB = 0.0F;
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);
B = (Npp8u) (nB * 255.0F);

```

- **NppStatus nppiLUVToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed LUV to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus nppiLUVToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed LUV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## BGRToLab

BGR to Lab color conversion.

This is how NPP converts gamma corrected BGR or RGB to Lab using the CIE Lab D65 white point with a Y luminance of 1.0. The computed values of the L component are in the range [0..100], a and b component values are in the range [-128..127]. The code uses `cbrtf()` the 32 bit floating point cube root math function.

```

// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert to XYZ
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nX = 0.412453F * nNormalizedR + 0.35758F * nNormalizedG + 0.180423F * nNormalizedB;
Npp32f nY = 0.212671F * nNormalizedR + 0.71516F * nNormalizedG + 0.072169F * nNormalizedB;
Npp32f nZ = 0.019334F * nNormalizedR + 0.119193F * nNormalizedG + 0.950227F * nNormalizedB;
Npp32f nL = cbrtf(nY);
Npp32f nA;
Npp32f nB;
Npp32f nfX = nX * 1.052128F; // / nCIE_LAB_D65_xn;
Npp32f nfY = nY;
Npp32f nfZ = nZ * 0.918482F; // / nCIE_LAB_D65_zn;
nfY = nL - 16.0F;
nL = 116.0F * nL - 16.0F;
nA = cbrtf(nfX) - 16.0F;
nA = 500.0F * (nA - nfY);
nB = cbrtf(nfZ) - 16.0F;
nB = 200.0F * (nfY - nB);
// Now scale Lab range
nL = nL * 255.0F * 0.01F; // / 100.0F
nA = nA + 128.0F;
nB = nB + 128.0F;
L = (Npp8u)nL;
a = (Npp8u)nA;
b = (Npp8u)nB;

```

- **NppStatus nppiBGRToLab\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed Lab color conversion.*

## LabToBGR

Lab to BGR color conversion.

This is how NPP converts Lab to gamma corrected BGR or RGB using the CIE Lab D65 white point with a Y luminance of 1.0. The code uses `powf()` the 32 bit floating point power math function.

```
// use CIE Lab chromaticity coordinates
#define nCIE_LAB_D65_xn 0.950455F
#define nCIE_LAB_D65_yn 1.0F
#define nCIE_LAB_D65_zn 1.088753F
// First convert Lab back to original range then to CIE XYZ
Npp32f nL = (Npp32f)L * 100.0F * 0.003921569F; // / 255.0F
Npp32f nA = (Npp32f)a - 128.0F;
Npp32f nB = (Npp32f)b - 128.0F;
Npp32f nP = (nL + 16.0F) * 0.008621F; // / 116.0F
Npp32f nNormalizedY = nP * nP * nP; // powf(nP, 3.0F);
Npp32f nNormalizedX = nCIE_LAB_D65_xn * powf((nP + nA * 0.002F), 3.0F); // / 500.0F
Npp32f nNormalizedZ = nCIE_LAB_D65_zn * powf((nP - nB * 0.005F), 3.0F); // / 200.0F
Npp32f nR = 3.240479F * nNormalizedX - 1.53715F * nNormalizedY - 0.498535F * nNormalizedZ;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = -0.969256F * nNormalizedX + 1.875991F * nNormalizedY + 0.041556F * nNormalizedZ;
if (nG > 1.0F)
    nG = 1.0F;
nB = 0.055648F * nNormalizedX - 0.204043F * nNormalizedY + 1.057311F * nNormalizedZ;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u)(nR * 255.0F);
G = (Npp8u)(nG * 255.0F);
B = (Npp8u)(nB * 255.0F);
```

- **NppStatus nppiLabToBGR\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed Lab to 3 channel 8-bit unsigned packed BGR color conversion.*

## RGBToYCC

RGB to PhotoYCC color conversion.

This is how NPP converts gamma corrected BGR or RGB to PhotoYCC. The computed Y, C1, C2 values are then quantized and converted to fit in the range [0..1] before expanding to 8 bits.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nY = 0.299F * nNormalizedR + 0.587F * nNormalizedG + 0.114F * nNormalizedB;
Npp32f nC1 = nNormalizedB - nY;
nC1 = 111.4F * 0.003921569F * nC1 + 156.0F * 0.003921569F;
Npp32f nC2 = nNormalizedR - nY;
nC2 = 135.64F * 0.003921569F * nC2 + 137.0F * 0.003921569F;
nY = 1.0F * 0.713267F * nY; // / 1.402F
Y = (Npp8u)(nY * 255.0F);
C1 = (Npp8u)(nC1 * 255.0F);
C2 = (Npp8u)(nC2 * 255.0F);
```

- **NppStatus nppiRGBToYCC\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YCC color conversion.*



- **NppStatus nppiRGBToYCC\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YCC with alpha color conversion.*

## YCCToRGB

PhotoYCC to RGB color conversion.

This is how NPP converts PhotoYCC to gamma corrected RGB or BGR.

```
Npp32f nNormalizedY = ((Npp32f)Y * 0.003921569F) * 1.3584F; // / 255.0F
Npp32f nNormalizedC1 = (((Npp32f)C1 * 0.003921569F) - 156.0F * 0.003921569F) * 2.2179F;
Npp32f nNormalizedC2 = (((Npp32f)C2 * 0.003921569F) - 137.0F * 0.003921569F) * 1.8215F;
Npp32f nR = nNormalizedY + nNormalizedC2;
if (nR > 1.0F)
    nR = 1.0F;
Npp32f nG = nNormalizedY - 0.194F * nNormalizedC1 - 0.509F * nNormalizedC2;
if (nG > 1.0F)
    nG = 1.0F;
Npp32f nB = nNormalizedY + nNormalizedC1;
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);
B = (Npp8u) (nB * 255.0F);
```

- **NppStatus nppiYCCToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed YCC to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus nppiYCCToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed YCC with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## RGBToHLS

RGB to HLS color conversion.

This is how NPP converts gamma corrected RGB or BGR to HLS. This code uses the fmaxf() and fminf() 32 bit floating point math functions.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
Npp32f nH;
// Lightness
Npp32f nMax = fmaxf(nNormalizedR, nNormalizedG);
nMax = fmaxf(nMax, nNormalizedB);
Npp32f nMin = fminf(nNormalizedR, nNormalizedG);
nMin = fminf(nMin, nNormalizedB);
Npp32f nL = (nMax + nMin) * 0.5F;
```

```

Npp32f nDivisor = nMax - nMin;
// Saturation
if (nDivisor == 0.0F) // achromatics case
{
    nS = 0.0F;
    nH = 0.0F;
}
else // chromatics case
{
    if (nL > 0.5F)
        nS = nDivisor / (1.0F - (nMax + nMin - 1.0F));
    else
        nS = nDivisor / (nMax + nMin);
}
// Hue
Npp32f nCr = (nMax - nNormalizedR) / nDivisor;
Npp32f nCg = (nMax - nNormalizedG) / nDivisor;
Npp32f nCb = (nMax - nNormalizedB) / nDivisor;
if (nNormalizedR == nMax)
    nH = nCb - nCg;
else if (nNormalizedG == nMax)
    nH = 2.0F + nCr - nCb;
else if (nNormalizedB == nMax)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.166667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u) (nH * 255.0F);
L = (Npp8u) (nL * 255.0F);
S = (Npp8u) (nS * 255.0F);

```

- [NppStatus nppiRGBToHLS\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HLS color conversion.*

- [NppStatus nppiRGBToHLS\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.*

## HLSToRGB

HLS to RGB color conversion.

This is how NPP converts HLS to gamma corrected RGB or BGR.

```

Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedL = (Npp32f)L * 0.003921569F;
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nM1;
Npp32f nM2;
Npp32f nR;
Npp32f nG;
Npp32f nB;
Npp32f nh = 0.0F;
if (nNormalizedL <= 0.5F)
    nM2 = nNormalizedL * (1.0F + nNormalizedS);
else
    nM2 = nNormalizedL + nNormalizedS - nNormalizedL * nNormalizedS;
nM1 = 2.0F * nNormalizedL - nM2;
if (nNormalizedS == 0.0F)

```

```

    nR = nG = nB = nNormalizedL;
else
{
    nh = nNormalizedH + 0.3333F;
    if (nh > 1.0F)
        nh -= 1.0F;
}
Npp32f nMDiff = nM2 - nM1;
if (0.6667F < nh)
    nR = nM1;
else
{
    if (nh < 0.1667F)
        nR = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nR = nM2;
    else
        nR = nM1 + nMDiff * ( 0.6667F - nh ) * 6.0F; // / 0.1667F
}
if (nR > 1.0F)
    nR = 1.0F;
nh = nNormalizedH;
if (0.6667F < nh)
    nG = nM1;
else
{
    if (nh < 0.1667F)
        nG = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nG = nM2;
    else
        nG = nM1 + nMDiff * (0.6667F - nh ) * 6.0F; // / 0.1667F
}
if (nG > 1.0F)
    nG = 1.0F;
nh = nNormalizedH - 0.3333F;
if (nh < 0.0F)
    nh += 1.0F;
if (0.6667F < nh)
    nB = nM1;
else
{
    if (nh < 0.1667F)
        nB = (nM1 + nMDiff * nh * 6.0F); // / 0.1667F
    else if (nh < 0.5F)
        nB = nM2;
    else
        nB = nM1 + nMDiff * (0.6667F - nh ) * 6.0F; // / 0.1667F
}
if (nB > 1.0F)
    nB = 1.0F;
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);
B = (Npp8u) (nB * 255.0F);

```

- **NppStatus nppiHLSToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned packed RGB color conversion.*

- **NppStatus nppiHLSToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## BGRToHLS

BGR to HLS color conversion.

- **NppStatus nppiBGRToHLS\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar HLS color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_AC4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned packed HLS color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_AP4C4R** (const **Npp8u** \*const pSrc[4], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar HLS color conversion.*
- **NppStatus nppiBGRToHLS\_8u\_AP4R** (const **Npp8u** \*const pSrc[4], int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.*

## HLSToBGR

HLS to BGR color conversion.

- **NppStatus nppiHLSToBGR\_8u\_C3P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned planar BGR color conversion.*
- **NppStatus nppiHLSToBGR\_8u\_AC4P4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.*

- **NppStatus nppiHLSToBGR\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned planar BGR color conversion.*
- **NppStatus nppiHLSToBGR\_8u\_AP4R** (const **Npp8u** \*const pSrc[4], int nSrcStep, **Npp8u** \*pDst[4], int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.*
- **NppStatus nppiHLSToBGR\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.*
- **NppStatus nppiHLSToBGR\_8u\_P3C3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned packed BGR color conversion.*
- **NppStatus nppiHLSToBGR\_8u\_AP4C4R** (const **Npp8u** \*const pSrc[4], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.*

## RGBToHSV

RGB to HSV color conversion.

This is how NPP converts gamma corrected RGB or BGR to HSV. This code uses the `fmaxf()` and `fminf()` 32 bit floating point math functions.

```
Npp32f nNormalizedR = (Npp32f)R * 0.003921569F; // / 255.0F
Npp32f nNormalizedG = (Npp32f)G * 0.003921569F;
Npp32f nNormalizedB = (Npp32f)B * 0.003921569F;
Npp32f nS;
Npp32f nH;
// Value
Npp32f nV = fmaxf(nNormalizedR, nNormalizedG);
nV = fmaxf(nV, nNormalizedB);
// Saturation
Npp32f nTemp = fminf(nNormalizedR, nNormalizedG);
nTemp = fminf(nTemp, nNormalizedB);
Npp32f nDivisor = nV - nTemp;
if (nV == 0.0F) // achromatics case
{
    nS = 0.0F;
    nH = 0.0F;
}
else // chromatics case
    nS = nDivisor / nV;
// Hue:
Npp32f nCr = (nV - nNormalizedR) / nDivisor;
Npp32f nCg = (nV - nNormalizedG) / nDivisor;
Npp32f nCb = (nV - nNormalizedB) / nDivisor;
if (nNormalizedR == nV)
    nH = nCb - nCg;
else if (nNormalizedG == nV)
    nH = 2.0F + nCr - nCb;
```

```

else if (nNormalizedB == nV)
    nH = 4.0F + nCg - nCr;
nH = nH * 0.166667F; // / 6.0F
if (nH < 0.0F)
    nH = nH + 1.0F;
H = (Npp8u) (nH * 255.0F);
S = (Npp8u) (nS * 255.0F);
V = (Npp8u) (nV * 255.0F);

```

- **NppStatus nppiRGBToHSV\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HSV color conversion.*

- **NppStatus nppiRGBToHSV\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HSV with alpha color conversion.*

## HSVToRGB

HSV to RGB color conversion.

This is how NPP converts HSV to gamma corrected RGB or BGR. This code uses the floorf() 32 bit floating point math function.

```

Npp32f nNormalizedH = (Npp32f)H * 0.003921569F; // / 255.0F
Npp32f nNormalizedS = (Npp32f)S * 0.003921569F;
Npp32f nNormalizedV = (Npp32f)V * 0.003921569F;
Npp32f nR;
Npp32f nG;
Npp32f nB;
if (nNormalizedS == 0.0F)
{
    nR = nG = nB = nNormalizedV;
}
else
{
    if (nNormalizedH == 1.0F)
        nNormalizedH = 0.0F;
    else
        nNormalizedH = nNormalizedH * 6.0F; // / 0.1667F
}
Npp32f nI = floorf(nNormalizedH);
Npp32f nF = nNormalizedH - nI;
Npp32f nM = nNormalizedV * (1.0F - nNormalizedS);
Npp32f nN = nNormalizedV * (1.0F - nNormalizedS * nF);
Npp32f nK = nNormalizedV * (1.0F - nNormalizedS * (1.0F - nF));
if (nI == 0.0F)
{ nR = nNormalizedV; nG = nK; nB = nM; }
else if (nI == 1.0F)
{ nR = nN; nG = nNormalizedV; nB = nM; }
else if (nI == 2.0F)
{ nR = nM; nG = nNormalizedV; nB = nK; }
else if (nI == 3.0F)
{ nR = nM; nG = nN; nB = nNormalizedV; }
else if (nI == 4.0F)
{ nR = nK; nG = nM; nB = nNormalizedV; }
else if (nI == 5.0F)
{ nR = nNormalizedV; nG = nM; nB = nN; }
R = (Npp8u) (nR * 255.0F);
G = (Npp8u) (nG * 255.0F);

```

$B = (\text{Npp8u})(nB * 255.0F);$

- **NppStatus nppiHSVToRGB\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed HSV to 3 channel 8-bit unsigned packed RGB color conversion.*
- **NppStatus nppiHSVToRGB\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed HSV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.*

## RGBToGray

RGB to CCIR601 Gray conversion.

Here is how NPP converts gamma corrected RGB to CCIR601 Gray.

$nGray = 0.299F * R + 0.587F * G + 0.114F * B;$

- **NppStatus nppiRGBToGray\_8u\_C3C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.*
- **NppStatus nppiRGBToGray\_8u\_AC4C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.*
- **NppStatus nppiRGBToGray\_16u\_C3C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.*
- **NppStatus nppiRGBToGray\_16u\_AC4C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.*
- **NppStatus nppiRGBToGray\_16s\_C3C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.*
- **NppStatus nppiRGBToGray\_16s\_AC4C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.*
- **NppStatus nppiRGBToGray\_32f\_C3C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.*
- **NppStatus nppiRGBToGray\_32f\_AC4C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.*

## ColorToGray

RGB Color to Gray conversion using user supplied conversion coefficients.

Here is how NPP converts gamma corrected RGB Color to Gray using user supplied conversion coefficients.

```
nGray = aCoeffs[0] * R + aCoeffs[1] * G + aCoeffs[2] * B;
```

- `NppStatus nppiColorToGray_8u_C3C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.*

- `NppStatus nppiColorToGray_8u_AC4C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.*

- `NppStatus nppiColorToGray_16u_C3C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.*

- `NppStatus nppiColorToGray_16u_AC4C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.*

- `NppStatus nppiColorToGray_16s_C3C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.*

- `NppStatus nppiColorToGray_16s_AC4C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.*

- `NppStatus nppiColorToGray_32f_C3C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.*

- `NppStatus nppiColorToGray_32f_AC4C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aCoeffs[3])

*4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.*

### 7.44.1 Detailed Description

Routines for converting between various image color models.



## 7.44.2 Function Documentation

### 7.44.2.1 NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_AC4C2R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.

images.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.44.2.2 NppStatus nppiBGRToCbYCr422\_709HDTV\_8u\_C3C2R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed CbYCr422\_709HDTV color conversion.

images.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.44.2.3 NppStatus nppiBGRToCbYCr422\_8u\_AC4C2R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

images.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.4 **NppStatus nppiBGRTToHLS\_8u\_AC4P4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.5 **NppStatus nppiBGRTToHLS\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.6 NppStatus nppiBGRToHLS\_8u\_AP4C4R (const Npp8u \*const pSrc[4], int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.7 NppStatus nppiBGRToHLS\_8u\_AP4R (const Npp8u \*const pSrc[4], int nSrcStep, Npp8u \*pDst[4], int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned planar BGR with alpha to 4 channel 8-bit unsigned planar HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.8 NppStatus nppiBGRToHLS\_8u\_C3P3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar HLS color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.9 NppStatus nppiBGRToHLS\_8u\_P3C3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned packed HLS color conversion.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.10 NppStatus nppiBGRToHLS\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar HLS color conversion.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.11 NppStatus nppiBGRToLab\_8u\_C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed Lab color conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.12 NppStatus nppiBGRToYCbCr411\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.13 NppStatus nppiBGRToYCbCr411\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr411 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.14 NppStatus nppiBGRToYCbCr420\_709CSC\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.15 **NppStatus nppiBGRToYCbCr420\_709CSC\_8u\_C3P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420\_709CSC color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.16 **NppStatus nppiBGRToYCbCr420\_709HDTV\_8u\_AC4P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420\_709HDTV color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.17 NppStatus nppiBGRToYCbCr420\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.18 NppStatus nppiBGRToYCbCr420\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr420 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.19 NppStatus nppiBGRToYCbCr422\_8u\_AC4C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 2 channel 8-bit unsigned packed YCrCb422 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.20 NppStatus nppiBGRToYCbCr422\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr422 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.21 NppStatus nppiBGRToYCbCr422\_8u\_C3C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 2 channel 8-bit unsigned packed YCrCb422 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.44.2.22 NppStatus nppiBGRToYCbCr422\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.23 NppStatus nppiBGRToYCbCr\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.24 NppStatus nppiBGRToYCbCr\_8u\_AC4P4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[4], int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YCbCr color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.25 NppStatus nppiBGRToYCbCr\_8u\_C3P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCbCr color conversion.  
 images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.26 NppStatus nppiBGRToYCrCb420\_709CSC\_8u\_AC4P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.  
 images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.27 NppStatus nppiBGRToYCrCb420\_709CSC\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420\_709CSC color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.28 NppStatus nppiBGRToYCrCb420\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.29 NppStatus nppiBGRToYCrCb420\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YCrCb420 color conversion.  
images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.30 **NppStatus nppiBGRToYUV420\_8u\_AC4P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 3 channel 8-bit unsigned planar YUV420 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.31 **NppStatus nppiBGRToYUV\_8u\_AC4P4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.32 NppStatus nppiBGRToYUV\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed BGR with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.33 NppStatus nppiBGRToYUV\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned planar YUV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.34 NppStatus nppiBGRToYUV\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed BGR to 3 channel 8-bit unsigned packed YUV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.35 NppStatus nppiBGRToYUV\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar BGR to 3 channel 8-bit unsigned planar YUV color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.36 NppStatus nppiCbYCr422ToBGR\_709HDTV\_8u\_C2C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned packed BGR\_709HDTV color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.37 NppStatus nppiCbYCr422ToBGR\_709HDTV\_8u\_C2C4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)**

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAval* 8-bit unsigned alpha constant.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.38 NppStatus nppiCbYCr422ToBGR\_8u\_C2C4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)**

2 channel 8-bit unsigned packed CbYCr422 to 4 channel 8-bit unsigned packed BGR color conversion with alpha.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nAval* 8-bit unsigned alpha constant.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.39 NppStatus nppiCbYCr422ToRGB\_8u\_C2C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 8-bit unsigned packed CbYCrC22 to 3 channel 8-bit unsigned packed RGB color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.40 NppStatus nppiColorToGray\_16s\_AC4C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])**

4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.41 NppStatus nppiColorToGray\_16s\_C3C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])**

3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.42 NppStatus nppiColorToGray\_16u\_AC4C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])**

4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.44.2.43 NppStatus nppiColorToGray\_16u\_C3C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aCoeffs*[3])**

3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.44 NppStatus nppiColorToGray\_32f\_AC4C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aCoeffs*[3])**

4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.45 NppStatus nppiColorToGray\_32f\_C3C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aCoeffs*[3])**

3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.46** `NppStatus nppiColorToGray_8u_AC4C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.47** `NppStatus nppiColorToGray_8u_C3C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aCoeffs[3])`

3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aCoeffs* fixed size array of constant floating point conversion coefficient values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.48 NppStatus nppiHLSToBGR\_8u\_AC4P4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[4], int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.49 NppStatus nppiHLSToBGR\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.50 NppStatus nppiHLSToBGR\_8u\_AP4C4R (const Npp8u \*const *pSrc*[4], int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned packed BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.51 NppStatus nppiHLSToBGR\_8u\_AP4R (const Npp8u \*const pSrc[4], int nSrcStep, Npp8u \*pDst[4], int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned planar HLS with alpha to 4 channel 8-bit unsigned planar BGR with alpha color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.52 NppStatus nppiHLSToBGR\_8u\_C3P3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned planar BGR color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.53 NppStatus nppiHLSToBGR\_8u\_P3C3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.54 NppStatus nppiHLSToBGR\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar HLS to 3 channel 8-bit unsigned planar BGR color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.55 NppStatus nppiHLSToRGB\_8u\_AC4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed HLS with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.56 NppStatus nppiHLSToRGB\_8u\_C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed HLS to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.57 NppStatus nppiHSVToRGB\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed HSV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.58 NppStatus nppiHSVToRGB\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed HSV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.59 NppStatus nppiLabToBGR\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed Lab to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.60 NppStatus nppiLUVToRGB\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed LUV with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.61 NppStatus nppiLUVToRGB\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed LUV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.62 NppStatus nppiRGBToCbYCr422\_8u\_C3C2R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed CbYCr422 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.63 NppStatus nppiRGBToCbYCr422Gamma\_8u\_C3C2R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB first gets forward gamma corrected then converted to 2 channel 8-bit unsigned packed CbYCr422 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.64 NppStatus nppiRGBToGray\_16s\_AC4C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit signed packed RGB with alpha to 1 channel 16-bit signed packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.65 NppStatus nppiRGBToGray\_16s\_C3C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 16-bit signed packed RGB to 1 channel 16-bit signed packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.44.2.66 NppStatus nppiRGBToGray\_16u\_AC4C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned packed RGB with alpha to 1 channel 16-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.67 NppStatus nppiRGBToGray\_16u\_C3C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 16-bit unsigned packed RGB to 1 channel 16-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.68 NppStatus nppiRGBToGray\_32f\_AC4C1R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 32-bit floating point packed RGB with alpha to 1 channel 32-bit floating point packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.69 NppStatus nppiRGBToGray\_32f\_C3C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 32-bit floating point packed RGB to 1 channel 32-bit floating point packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.70 NppStatus nppiRGBToGray\_8u\_AC4C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed RGB with alpha to 1 channel 8-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.71 NppStatus nppiRGBToGray\_8u\_C3C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed RGB to 1 channel 8-bit unsigned packed Gray conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.72 NppStatus nppiRGBToHLS\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HLS with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.73 NppStatus nppiRGBToHLS\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HLS color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.74 NppStatus nppiRGBToHSV\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed HSV with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.75 NppStatus nppiRGBToHSV\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed HSV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.76 NppStatus nppiRGBToLUV\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed LUV with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.77 NppStatus nppiRGBToLUV\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed LUV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.78 NppStatus nppiRGBToXYZ\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed XYZ with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.79 NppStatus nppiRGBToXYZ\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed XYZ color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.80 NppStatus nppiRGBToYCbCr420\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr420 color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.81 **NppStatus nppiRGBToYCbCr422\_8u\_C3C2R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion. images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.82 **NppStatus nppiRGBToYCbCr422\_8u\_C3P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YCbCr422 color conversion. images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.83 **NppStatus nppiRGBToYCbCr422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCbCr422 color conversion. images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.84** `NppStatus nppiRGBToYCbCr_8u_AC4P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCbCr color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.85** `NppStatus nppiRGBToYCbCr_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed RGB with alpha to 4 channel unsigned 8-bit packed YCbCr with alpha color conversion, not affecting alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.86** `NppStatus nppiRGBToYCbCr_8u_C3P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit planar YCbCr color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.87** `NppStatus nppiRGBToYCbCr_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel unsigned 8-bit packed YCbCr color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.88** `NppStatus nppiRGBToYCbCr_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel planar 8-bit unsigned RGB to 3 channel planar 8-bit YCbCr color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.44.2.89 NppStatus nppiRGBToYCC\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YCC with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.90 NppStatus nppiRGBToYCC\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YCC color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.91 NppStatus nppiRGBToYCrCb420\_8u\_AC4P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int rDstStep[3], NppiSize oSizeROI)**

4 channel 8-bit unsigned packed RGB with alpha to 3 channel 8-bit unsigned planar YCrCb420 color conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.92 **NppStatus nppiRGBToYCrCb422\_8u\_C3C2R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.93 **NppStatus nppiRGBToYCrCb422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar RGB to 2 channel 8-bit unsigned packed YCrCb422 color conversion. images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.94 **NppStatus nppiRGBToYUV420\_8u\_C3P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV420 color conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.95** `NppStatus nppiRGBToYUV420_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV420 color conversion. images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.96** `NppStatus nppiRGBToYUV422_8u_C3C2R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 2 channel 8-bit unsigned packed YUV422 color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.97** `NppStatus nppiRGBToYUV422_8u_C3P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV422 color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.98 NppStatus nppiRGBToYUV422\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int rDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV422 color conversion. images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.99 NppStatus nppiRGBToYUV\_8u\_AC4P4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst[4], int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned planar YUV color conversion with alpha. images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.100 NppStatus nppiRGBToYUV\_8u\_AC4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)

4 channel 8-bit unsigned packed RGB with alpha to 4 channel 8-bit unsigned packed YUV color conversion with alpha, not affecting alpha. images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.101 NppStatus nppiRGBToYUV\_8u\_C3P3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned planar YUV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.102 NppStatus nppiRGBToYUV\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed RGB to 3 channel 8-bit unsigned packed YUV color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.103 NppStatus nppiRGBToYUV\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \* pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar RGB to 3 channel 8-bit unsigned planar YUV color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.104 NppStatus nppiXYZToRGB\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned packed XYZ with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.105 NppStatus nppiXYZToRGB\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed XYZ to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.106 NppStatus nppiYCbCr411ToBGR\_8u\_P3C3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.107 NppStatus nppiYCbCr411ToBGR\_8u\_P3C4R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)**

3 channel 8-bit unsigned planar YCbCr411 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.108 NppStatus nppiYCbCr420ToBGR\_709CSC\_8u\_P3C3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.109 **NppStatus nppiYCbCr420ToBGR\_709HDTV\_8u\_P3C4R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nAval*)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR\_709HDTV color conversion with constant alpha.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.110 **NppStatus nppiYCbCr420ToBGR\_8u\_P3C3R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed BGR color conversion.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.111 **NppStatus nppiYCbCr420ToBGR\_8u\_P3C4R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nAval*)

3 channel 8-bit unsigned planar YCbCr420 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).



*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.112** `NppStatus nppiYCbCr420ToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*rSrcStep* [Source-Planar-Image Line Step Array](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.113** `NppStatus nppiYCbCr422ToBGR_8u_C2C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed BGR color conversion. images.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.114** `NppStatus nppiYCbCr422ToBGR_8u_C2C4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

2 channel 8-bit unsigned packed YCrCb422 to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.115** `NppStatus nppiYCbCr422ToBGR_8u_P3C3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed BGR color conversion. images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.116** `NppStatus nppiYCbCr422ToRGB_8u_C2C3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.117 NppStatus nppiYCbCr422ToRGB\_8u\_C2P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)**

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar RGB color conversion. images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.118 NppStatus nppiYCbCr422ToRGB\_8u\_P3C3R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.119 NppStatus nppiYCbCrToBGR\_709CSC\_8u\_P3C3R (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR\_709CSC color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.120 **NppStatus nppiYCbCrToBGR\_709CSC\_8u\_P3C4R** (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nAval*)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR\_709CSC color conversion with constant alpha.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.121 **NppStatus nppiYCbCrToBGR\_8u\_P3C3R** (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed BGR color conversion.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.122 **NppStatus nppiYCbCrToBGR\_8u\_P3C4R** (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nAval*)

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed BGR color conversion with constant alpha.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nAval* 8-bit unsigned alpha constant.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.123** `NppStatus nppiYCbCrToRGB_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned packed YCbCr with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.124** `NppStatus nppiYCbCrToRGB_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned packed YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.125** `NppStatus nppiYCbCrToRGB_8u_P3C3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.126** `NppStatus nppiYCbCrToRGB_8u_P3C4R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, Npp8u nAval)`

3 channel 8-bit unsigned planar YCbCr to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.127** `NppStatus nppiYCbCrToRGB_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.128 NppStatus nppiYCCToRGB\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned packed YCC with alpha to 4 channel 8-bit unsigned packed RGB with alpha color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.129 NppStatus nppiYCCToRGB\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed YCC to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.130 NppStatus nppiYCrCb420ToRGB\_8u\_P3C4R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nAval*)**

3 channel 8-bit unsigned planar YCrCb420 to 4 channel 8-bit unsigned packed RGB color conversion with constant alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nAval* 8-bit unsigned alpha constant.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.131 **NppStatus nppiYCrCb422ToRGB\_8u\_C2C3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned packed RGB color conversion. images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.132 **NppStatus nppiYCrCb422ToRGB\_8u\_C2P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar RGB color conversion. images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.133 **NppStatus nppiYUV420ToBGR\_8u\_P3C3R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed BGR color conversion.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.44.2.134 NppStatus nppiYUV420ToRGB\_8u\_P3AC4R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV420 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.135 NppStatus nppiYUV420ToRGB\_8u\_P3C3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.136 NppStatus nppiYUV420ToRGB\_8u\_P3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV420 to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.137 NppStatus nppiYUV422ToRGB\_8u\_C2C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

2 channel 8-bit unsigned packed YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.138 NppStatus nppiYUV422ToRGB\_8u\_P3AC4R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned planar YUV422 to 4 channel 8-bit unsigned packed RGB color conversion with alpha.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.139 NppStatus nppiYUV422ToRGB\_8u\_P3C3R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.140 NppStatus nppiYUV422ToRGB\_8u\_P3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar YUV422 to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.141 NppStatus nppiYUVToBGR\_8u\_AC4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed BGR color conversion with alpha, not affecting alpha.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.142 NppStatus nppiYUVToBGR\_8u\_C3R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed BGR color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.44.2.143 **NppStatus nppiYUVToBGR\_8u\_P3C3R** (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed BGR color conversion.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.144 **NppStatus nppiYUVToBGR\_8u\_P3R** (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \**pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar BGR color conversion.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.44.2.145 **NppStatus nppiYUVToRGB\_8u\_AC4R** (const Npp8u \**pSrc*, int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit packed YUV with alpha to 4 channel 8-bit unsigned packed RGB color conversion with alpha, not affecting alpha.

images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.44.2.146 NppStatus nppiYUVToRGB\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned packed YUV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.147 NppStatus nppiYUVToRGB\_8u\_P3C3R (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned packed RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.44.2.148 NppStatus nppiYUVToRGB\_8u\_P3R (const Npp8u \*const *pSrc*[3], int *nSrcStep*, Npp8u \* *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned planar YUV to 3 channel 8-bit unsigned planar RGB color conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.45 Color Sampling Format Conversion

Routines for converting between various image color sampling formats.

### YCbCr420ToYCbCr411

YCbCr420 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr420ToYCbCr411\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*
- **NppStatus** **nppiYCbCr420ToYCbCr411\_8u\_P2P3R** (const **Npp8u** \*pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

### YCbCr422ToYCbCr422

YCbCr422 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr422\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*
- **NppStatus** **nppiYCbCr422\_8u\_P3C2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

### YCbCr422ToYCrCb422

YCbCr422 to YCrCb422 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCrCb422\_8u\_C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCrCb422\_8u\_P3C2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.*

**YCbCr422ToCbYCr422**

YCbCr422 to CbYCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToCbYCr422\_8u\_C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.*

**CbYCr422ToYCbCr411**

CbYCr422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr411\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

**YCbCr422ToYCbCr420**

YCbCr422 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int nDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus** **nppiYCbCr422ToYCbCr420\_8u\_C2P2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

**YCrCb420ToYCbCr422**

YCrCb420 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCrCb420ToYCbCr422\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCrCb420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

- **NppStatus** **nppiYCrCb420ToYCbCr422\_8u\_P3C2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

## YCbCr422ToYCrCb420

YCbCr422 to YCrCb420 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCrCb420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

## YCbCr422ToYCbCr411

YCbCr422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

- **NppStatus** **nppiYCbCr422ToYCbCr411\_8u\_C2P2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

## YCrCb422ToYCbCr422

YCrCb422 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCrCb422ToYCbCr422\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)



*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

### YCrCb422ToYCbCr420

YCrCb422 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCrCb422ToYCbCr420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

### YCrCb422ToYCbCr411

YCrCb422 to YCbCr411 sampling format conversion.

- **NppStatus** **nppiYCrCb422ToYCbCr411\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

### CbYCr422ToYCbCr422

CbYCr422 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr422\_8u\_C2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

- **NppStatus** **nppiCbYCr422ToYCbCr422\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

### CbYCr422ToYCbCr420

CbYCr422 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCbCr420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

- **NppStatus** **nppiCbYCr422ToYCbCr420\_8u\_C2P2R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

## CbYCr422ToYCrCb420

CbYCr422 to YCrCb420 sampling format conversion.

- **NppStatus** **nppiCbYCr422ToYCrCb420\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

## YCbCr420ToYCbCr420

YCbCr420 to YCbCr420 sampling format conversion.

- **NppStatus** **nppiYCbCr420\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

- **NppStatus** **nppiYCbCr420\_8u\_P2P3R** (const **Npp8u** \*const pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

## YCbCr420ToYCbCr422

YCbCr420 to YCbCr422 sampling format conversion.

- **NppStatus** **nppiYCbCr420ToYCbCr422\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int nDstStep[3], **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

- **NppStatus** **nppiYCbCr420ToYCbCr422\_8u\_P2P3R** (const **Npp8u** \*pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*

- **NppStatus** **nppiYCbCr420ToYCbCr422\_8u\_P2C2R** (const **Npp8u** \*pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

## YCbCr420ToCbYCr422

YCbCr420 to CbYCr422 sampling format conversion.

- `NppStatus nppiYCbCr420ToCbYCr422_8u_P2C2R` (const `Npp8u` \*pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.*

## YCbCr420ToYCrCb420

YCbCr420 to YCrCb420 sampling format conversion.

- `NppStatus nppiYCbCr420ToYCrCb420_8u_P2P3R` (const `Npp8u` \*pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

## YCrCb420ToCbYCr422

YCrCb420 to CbYCr422 sampling format conversion.

- `NppStatus nppiYCrCb420ToCbYCr422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.*

## YCrCb420ToYCbCr420

YCrCb420 to YCbCr420 sampling format conversion.

- `NppStatus nppiYCrCb420ToYCbCr420_8u_P3P2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDstY, int nDstYStep, `Npp8u` \*pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

## YCrCb420ToYCbCr411

YCrCb420 to YCbCr411 sampling format conversion.

- `NppStatus nppiYCrCb420ToYCbCr411_8u_P3P2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDstY, int nDstYStep, `Npp8u` \*pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

## YCbCr411ToYCbCr411

YCbCr411 to YCbCr411 sampling format conversion.

- `NppStatus nppiYCbCr411_8u_P3P2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDstY, int nDstYStep, `Npp8u` \*pDstCbCr, int nDstCbCrStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*
- `NppStatus nppiYCbCr411_8u_P2P3R` (const `Npp8u` \*pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.*

## YCbCr411ToYCbCr422

YCbCr411 to YCbCr422 sampling format conversion.

- `NppStatus nppiYCbCr411ToYCbCr422_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr422_8u_P2P3R` (const `Npp8u` \*const pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst[3], int rDstStep[3], `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCbCr422_8u_P2C2R` (const `Npp8u` \*pSrcY, int nSrcYStep, const `Npp8u` \*pSrcCbCr, int nSrcCbCrStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.*

## YCbCr411ToYCrCb422

YCbCr411 to YCrCb422 sampling format conversion.

- `NppStatus nppiYCbCr411ToYCrCb422_8u_P3R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst[3], int nDstStep[3], `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb422 sampling format conversion.*
- `NppStatus nppiYCbCr411ToYCrCb422_8u_P3C2R` (const `Npp8u` \*const pSrc[3], int rSrcStep[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.*

## YCbCr411ToYCbCr420

YCbCr411 to YCbCr420 sampling format conversion.

- **NppStatus nppiYCbCr411ToYCbCr420\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDst[3], int nDstStep[3], **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus nppiYCbCr411ToYCbCr420\_8u\_P3P2R** (const **Npp8u** \*const pSrc[3], int rSrcStep[3], **Npp8u** \*pDstY, int nDstYStep, **Npp8u** \*pDstCbCr, int nDstCbCrStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*
- **NppStatus nppiYCbCr411ToYCbCr420\_8u\_P2P3R** (const **Npp8u** \*pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.*

## YCbCr411ToYCrCb420

YCbCr411 to YCrCb420 sampling format conversion.

- **NppStatus nppiYCbCr411ToYCrCb420\_8u\_P2P3R** (const **Npp8u** \*pSrcY, int nSrcYStep, const **Npp8u** \*pSrcCbCr, int nSrcCbCrStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.*

### 7.45.1 Detailed Description

Routines for converting between various image color sampling formats.

### 7.45.2 Function Documentation

#### 7.45.2.1 **NppStatus nppiCbYCr422ToYCbCr411\_8u\_C2P3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst[3], int rDstStep[3], **NppiSize** oSizeROI)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.2 **NppStatus nppiCbYCr422ToYCbCr420\_8u\_C2P2R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDstY*, int *nDstYStep*, Npp8u \* *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.3 **NppStatus nppiCbYCr422ToYCbCr420\_8u\_C2P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.4 NppStatus nppiCbYCr422ToYCbCr422\_8u\_C2P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)**

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.5 NppStatus nppiCbYCr422ToYCbCr422\_8u\_C2R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

2 channel 8-bit unsigned packed CbYCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.6 NppStatus nppiCbYCr422ToYCrCb420\_8u\_C2P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)**

2 channel 8-bit unsigned packed CbYCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.7** `NppStatus nppiYCbCr411_8u_P2P3R (const Npp8u *pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.8** `NppStatus nppiYCbCr411_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



### 7.45.2.9 NppStatus nppiYCbCr411ToYCbCr420\_8u\_P2P3R (const Npp8u \* *pSrcY*, int *nSrcYStep*, const Npp8u \* *pSrcCbCr*, int *nSrcCbCrStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

#### Parameters:

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.45.2.10 NppStatus nppiYCbCr411ToYCbCr420\_8u\_P3P2R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDstY*, int *nDstYStep*, Npp8u \* *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

#### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.45.2.11 NppStatus nppiYCbCr411ToYCbCr420\_8u\_P3R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*[3], int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.12** `NppStatus nppiYCbCr411ToYCbCr422_8u_P2C2R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.13** `NppStatus nppiYCbCr411ToYCbCr422_8u_P2P3R (const Npp8u * const pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.14 **NppStatus nppiYCbCr411ToYCbCr422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.15 **NppStatus nppiYCbCr411ToYCbCr422\_8u\_P3R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*[3], int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.16 **NppStatus nppiYCbCr411ToYCrCb420\_8u\_P2P3R** (const Npp8u \**pSrcY*, int *nSrcYStep*, const Npp8u \**pSrcCbCr*, int *nSrcCbCrStep*, Npp8u \**pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

##### Parameters:

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.

*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.17 NppStatus nppiYCbCr411ToYCrCb422\_8u\_P3C2R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr411 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.18 NppStatus nppiYCbCr411ToYCrCb422\_8u\_P3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst[3], int nDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr411 to 3 channel 8-bit unsigned planar YCrCb422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.19** `NppStatus nppiYCbCr420_8u_P2P3R (const Npp8u *const pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.20** `NppStatus nppiYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.21** `NppStatus nppiYCbCr420ToCbYCr422_8u_P2C2R (const Npp8u *pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.22** `NppStatus nppiYCbCr420ToYCbCr411_8u_P2P3R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.23** `NppStatus nppiYCbCr420ToYCbCr411_8u_P3P2R (const Npp8u * const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.

*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.24** `NppStatus nppiYCbCr420ToYCbCr422_8u_P2C2R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.25** `NppStatus nppiYCbCr420ToYCbCr422_8u_P2P3R (const Npp8u * pSrcY, int nSrcYStep, const Npp8u * pSrcCbCr, int nSrcCbCrStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

**Parameters:**

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.26 `NppStatus nppiYCbCr420ToYCbCr422_8u_P3R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDst[3], int nDstStep[3], NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.27 `NppStatus nppiYCbCr420ToYCrCb420_8u_P2P3R (const Npp8u *pSrcY, int nSrcYStep, const Npp8u *pSrcCbCr, int nSrcCbCrStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned planar YCbCr420 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

##### Parameters:

*pSrcY* Source-Planar-Image Pointer.  
*nSrcYStep* Source-Planar-Image Line Step.  
*pSrcCbCr* Source-Planar-Image Pointer.  
*nSrcCbCrStep* Source-Planar-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.28 `NppStatus nppiYCbCr422_8u_C2P3R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.29 **NppStatus nppiYCbCr422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.30 **NppStatus nppiYCbCr422ToCbYCr422\_8u\_C2R** (const Npp8u \**pSrc*, int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.31 NppStatus nppiYCbCr422ToYCbCr411\_8u\_C2P2R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDstY*, int *nDstYStep*, Npp8u \* *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)**

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.32 NppStatus nppiYCbCr422ToYCbCr411\_8u\_C2P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)**

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.33 NppStatus nppiYCbCr422ToYCbCr411\_8u\_P3P2R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDstY*, int *nDstYStep*, Npp8u \* *pDstCbCr*, int *nDstCbCrStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.34 NppStatus nppiYCbCr422ToYCbCr411\_8u\_P3R (const Npp8u \*const pSrc[3], int rSrcStep[3], Npp8u \*pDst[3], int rDstStep[3], NppiSize oSizeROI)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.35 NppStatus nppiYCbCr422ToYCbCr420\_8u\_C2P2R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDstY, int nDstYStep, Npp8u \*pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.

*nDstCbCrStep* Destination-Planar-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.36** `NppStatus nppiYCbCr422ToYCbCr420_8u_C2P3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst[3], int rDstStep[3], NppiSize oSizeROI)`

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Planar-Image Pointer Array.

*rDstStep* Destination-Planar-Image Line Step Array.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.37** `NppStatus nppiYCbCr422ToYCbCr420_8u_P3P2R (const Npp8u * const pSrc[3], int rSrcStep[3], Npp8u * pDstY, int nDstYStep, Npp8u * pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.

*rSrcStep* Source-Planar-Image Line Step Array.

*pDstY* Destination-Planar-Image Pointer.

*nDstYStep* Destination-Planar-Image Line Step.

*pDstCbCr* Destination-Planar-Image Pointer.

*nDstCbCrStep* Destination-Planar-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.38 NppStatus nppiYCbCr422ToYCbCr420\_8u\_P3R (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*[3], int *nDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.39 NppStatus nppiYCbCr422ToYCrCb420\_8u\_C2P3R (const Npp8u \**pSrc*, int *nSrcStep*, Npp8u \**pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 3 channel 8-bit unsigned planar YCrCb420 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.40 NppStatus nppiYCbCr422ToYCrCb422\_8u\_C2R (const Npp8u \**pSrc*, int *nSrcStep*, Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.41 **NppStatus nppiYCbCr422ToYCrCb422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCbCr422 to 2 channel 8-bit unsigned packed YCrCb422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.42 **NppStatus nppiYCrCb420ToCbYCr422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed CbYCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.43** `NppStatus nppiYCrCb420ToYCbCr411_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.45.2.44** `NppStatus nppiYCrCb420ToYCbCr420_8u_P3P2R (const Npp8u *const pSrc[3], int rSrcStep[3], Npp8u *pDstY, int nDstYStep, Npp8u *pDstCbCr, int nDstCbCrStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDstY* Destination-Planar-Image Pointer.  
*nDstYStep* Destination-Planar-Image Line Step.  
*pDstCbCr* Destination-Planar-Image Pointer.  
*nDstCbCrStep* Destination-Planar-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.45 **NppStatus nppiYCrCb420ToYCbCr422\_8u\_P3C2R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCrCb420 to 2 channel 8-bit unsigned packed YCbCr422 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.45.2.46 **NppStatus nppiYCrCb420ToYCbCr422\_8u\_P3R** (const Npp8u \*const *pSrc*[3], int *rSrcStep*[3], Npp8u \**pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar YCrCb420 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Planar-Image Pointer Array.  
*rSrcStep* Source-Planar-Image Line Step Array.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.45.2.47 **NppStatus nppiYCrCb422ToYCbCr411\_8u\_C2P3R** (const Npp8u \**pSrc*, int *nSrcStep*, Npp8u \**pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr411 sampling format conversion.

images.

##### Parameters:

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.48 NppStatus nppiYCrCb422ToYCbCr420\_8u\_C2P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr420 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.45.2.49 NppStatus nppiYCrCb422ToYCbCr422\_8u\_C2P3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*[3], int *rDstStep*[3], NppiSize *oSizeROI*)

2 channel 8-bit unsigned packed YCrCb422 to 3 channel 8-bit unsigned planar YCbCr422 sampling format conversion.

images.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Planar-Image Pointer Array.  
*rDstStep* Destination-Planar-Image Line Step Array.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.46 Color Gamma Correction

Routines for correcting image color gamma.

### GammaFwd

Forward gamma correction.

- **NppStatus nppiGammaFwd\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed color not in place forward gamma correction.*
- **NppStatus nppiGammaFwd\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed color in place forward gamma correction.*
- **NppStatus nppiGammaFwd\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed color with alpha not in place forward gamma correction.*
- **NppStatus nppiGammaFwd\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed color with alpha in place forward gamma correction.*
- **NppStatus nppiGammaFwd\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar color not in place forward gamma correction.*
- **NppStatus nppiGammaFwd\_8u\_IP3R** (**Npp8u** \*const pSrcDst[3], int nSrcDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned planar color in place forward gamma correction.*

### GammaInv

Inverse gamma correction.

- **NppStatus nppiGammaInv\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed color not in place inverse gamma correction.*
- **NppStatus nppiGammaInv\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned packed color in place inverse gamma correction.*
- **NppStatus nppiGammaInv\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed color with alpha not in place inverse gamma correction.*
- **NppStatus nppiGammaInv\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned packed color with alpha in place inverse gamma correction.*

- **NppStatus nppiGammaInv\_8u\_P3R** (const **Npp8u** \*const pSrc[3], int nSrcStep, **Npp8u** \*pDst[3], int nDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar color not in place inverse gamma correction.*

- **NppStatus nppiGammaInv\_8u\_IP3R** (**Npp8u** \*const pSrcDst[3], int nSrcDstStep, **NppiSize** oSizeROI)

*3 channel 8-bit unsigned planar color in place inverse gamma correction.*

### 7.46.1 Detailed Description

Routines for correcting image color gamma.

### 7.46.2 Function Documentation

#### 7.46.2.1 **NppStatus nppiGammaFwd\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed color with alpha in place forward gamma correction.

##### Parameters:

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.46.2.2 **NppStatus nppiGammaFwd\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

4 channel 8-bit unsigned packed color with alpha not in place forward gamma correction.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.46.2.3 NppStatus nppiGammaFwd\_8u\_C3IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color in place forward gamma correction.

#### Parameters:

*pSrcDst* in place packed pixel image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.46.2.4 NppStatus nppiGammaFwd\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color not in place forward gamma correction.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.46.2.5 NppStatus nppiGammaFwd\_8u\_IP3R (Npp8u \*const *pSrcDst*[3], int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar color in place forward gamma correction.

#### Parameters:

*pSrcDst* in place planar pixel format image pointer array.

*nSrcDstStep* in place planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.6 NppStatus nppiGammaFwd\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*pDst[3], int nDstStep, NppiSize oSizeROI)**

3 channel 8-bit unsigned planar color not in place forward gamma correction.

**Parameters:**

*pSrc* source planar pixel format image pointer array.  
*nSrcStep* source planar pixel format image line step.  
*pDst* destination planar pixel format image pointer array.  
*nDstStep* destination planar pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.7 NppStatus nppiGammaInv\_8u\_AC4IR (Npp8u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed color with alpha in place inverse gamma correction.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.8 NppStatus nppiGammaInv\_8u\_AC4R (const Npp8u \*pSrc, int nSrcStep, Npp8u \*pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 8-bit unsigned packed color with alpha not in place inverse gamma correction.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.46.2.9 NppStatus nppiGammaInv\_8u\_C3IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color in place inverse gamma correction.

##### Parameters:

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.46.2.10 NppStatus nppiGammaInv\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned packed color not in place inverse gamma correction.

##### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.46.2.11 NppStatus nppiGammaInv\_8u\_IP3R (Npp8u \*const *pSrcDst*[3], int *nSrcDstStep*, NppiSize *oSizeROI*)

3 channel 8-bit unsigned planar color in place inverse gamma correction.

##### Parameters:

*pSrcDst* in place planar pixel format image pointer array.  
*nSrcDstStep* in place planar pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.46.2.12** `NppStatus nppiGammaInv_8u_P3R (const Npp8u *const pSrc[3], int nSrcStep, Npp8u *pDst[3], int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned planar color not in place inverse gamma correction.

**Parameters:**

*pSrc* source planar pixel format image pointer array.

*nSrcStep* source planar pixel format image line step.

*pDst* destination planar pixel format image pointer array.

*nDstStep* destination planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.47 Complement Color Key

Routines for performing complement color key replacement.

### CompColorKey

Complement color key replacement.

- **NppStatus nppiCompColorKey\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst)  
*1 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.*
- **NppStatus nppiCompColorKey\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[3])  
*3 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.*
- **NppStatus nppiCompColorKey\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[4])  
*4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.*
- **NppStatus nppiAlphaCompColorKey\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[4], **NppiAlphaOp** nppAlphaOp)  
*4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2 with alpha blending.*

### 7.47.1 Detailed Description

Routines for performing complement color key replacement.

### 7.47.2 Function Documentation

**7.47.2.1 NppStatus nppiAlphaCompColorKey\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, **Npp8u** nAlpha1, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** nAlpha2, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp8u** nColorKeyConst[4], **NppiAlphaOp** nppAlphaOp)

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2 with alpha blending.

#### Parameters:

- pSrc1** source1 packed pixel format image pointer.
- nSrc1Step** source1 packed pixel format image line step.
- nAlpha1** source1 image alpha opacity (0 - max channel pixel value).
- pSrc2** source2 packed pixel format image pointer.



*nSrc2Step* source2 packed pixel format image line step.  
*nAlpha2* source2 image alpha opacity (0 - max channel pixel value).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nColorKeyConst* color key constant array  
*nppAlphaOp* NppiAlphaOp alpha compositing operation selector (excluding premul ops).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.47.2.2 NppStatus nppiCompColorKey\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nColorKeyConst*)

1 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

**Parameters:**

*pSrc1* source1 packed pixel format image pointer.  
*nSrc1Step* source1 packed pixel format image line step.  
*pSrc2* source2 packed pixel format image pointer.  
*nSrc2Step* source2 packed pixel format image line step.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nColorKeyConst* color key constant

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.47.2.3 NppStatus nppiCompColorKey\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp8u *nColorKeyConst*[3])

3 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

**Parameters:**

*pSrc1* source1 packed pixel format image pointer.  
*nSrc1Step* source1 packed pixel format image line step.  
*pSrc2* source2 packed pixel format image pointer.  
*nSrc2Step* source2 packed pixel format image line step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nColorKeyConst* color key constant array

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.47.2.4** `NppStatus nppiCompColorKey_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp8u nColorKeyConst[4])`

4 channel 8-bit unsigned packed color complement color key replacement of source image 1 by source image 2.

**Parameters:**

*pSrc1* source1 packed pixel format image pointer.  
*nSrc1Step* source1 packed pixel format image line step.  
*pSrc2* source2 packed pixel format image pointer.  
*nSrc2Step* source2 packed pixel format image line step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nColorKeyConst* color key constant array

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.48 Color Processing

Routines for performing image color manipulation.

### ColorTwist

Perform color twist pixel processing.

Color twist consists of applying the following formula to each image pixel using coefficients from the user supplied color twist host matrix array as follows where `dst[x]` and `src[x]` represent destination pixel and source pixel channel or plane `x`.

$$\begin{aligned} \text{dst}[0] &= \text{aTwist}[0][0] * \text{src}[0] + \text{aTwist}[0][1] * \text{src}[1] + \text{aTwist}[0][2] * \text{src}[2] + \text{aTwist}[0][3] \\ \text{dst}[1] &= \text{aTwist}[1][0] * \text{src}[0] + \text{aTwist}[1][1] * \text{src}[1] + \text{aTwist}[1][2] * \text{src}[2] + \text{aTwist}[1][3] \\ \text{dst}[2] &= \text{aTwist}[2][0] * \text{src}[0] + \text{aTwist}[2][1] * \text{src}[1] + \text{aTwist}[2][2] * \text{src}[2] + \text{aTwist}[2][3] \end{aligned}$$

- `NppStatus nppiColorTwist32f_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*3 channel 8-bit unsigned color twist.*
- `NppStatus nppiColorTwist32f_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*3 channel 8-bit unsigned in place color twist.*
- `NppStatus nppiColorTwist32f_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*4 channel 8-bit unsigned color twist, with alpha copy.*
- `NppStatus nppiColorTwist32f_8u_C4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*4 channel 8-bit unsigned in place color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*4 channel 8-bit unsigned color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*4 channel 8-bit unsigned in place color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_8u_P3R` (const `Npp8u` \*const pSrc[3], int nSrcStep, `Npp8u` \*const pDst[3], int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*3 channel 8-bit unsigned planar color twist.*
- `NppStatus nppiColorTwist32f_8u_IP3R` (`Npp8u` \*const pSrcDst[3], int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*3 channel 8-bit unsigned planar in place color twist.*
- `NppStatus nppiColorTwist32f_8s_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*3 channel 8-bit signed color twist.*

- `NppStatus nppiColorTwist32f_8s_C3IR (Npp8s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*3 channel 8-bit signed in place color twist.*
- `NppStatus nppiColorTwist32f_8s_C4R (const Npp8s *pSrc, int nSrcStep, Npp8s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*4 channel 8-bit signed color twist, with alpha copy.*
- `NppStatus nppiColorTwist32f_8s_C4IR (Npp8s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*4 channel 8-bit signed in place color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_8s_AC4R (const Npp8s *pSrc, int nSrcStep, Npp8s *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*4 channel 8-bit signed color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_8s_AC4IR (Npp8s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*4 channel 8-bit signed in place color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_8s_P3R (const Npp8s *const pSrc[3], int nSrcStep, Npp8s *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*3 channel 8-bit signed planar color twist.*
- `NppStatus nppiColorTwist32f_8s_IP3R (Npp8s *const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*3 channel 8-bit signed planar in place color twist.*
- `NppStatus nppiColorTwist32f_16u_C3R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*3 channel 16-bit unsigned color twist.*
- `NppStatus nppiColorTwist32f_16u_C3IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*3 channel 16-bit unsigned in place color twist.*
- `NppStatus nppiColorTwist32f_16u_AC4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*4 channel 16-bit unsigned color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_16u_AC4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*4 channel 16-bit unsigned in place color twist, not affecting Alpha.*
- `NppStatus nppiColorTwist32f_16u_P3R (const Npp16u *const pSrc[3], int nSrcStep, Npp16u *const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`  
*3 channel 16-bit unsigned planar color twist.*
- `NppStatus nppiColorTwist32f_16u_IP3R (Npp16u *const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

*3 channel 16-bit unsigned planar in place color twist.*

- `NppStatus nppiColorTwist32f_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*3 channel 16-bit signed color twist.*

- `NppStatus nppiColorTwist32f_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*3 channel 16-bit signed in place color twist.*

- `NppStatus nppiColorTwist32f_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*4 channel 16-bit signed color twist, not affecting Alpha.*

- `NppStatus nppiColorTwist32f_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*4 channel 16-bit signed in place color twist, not affecting Alpha.*

- `NppStatus nppiColorTwist32f_16s_P3R` (const `Npp16s` \*const pSrc[3], int nSrcStep, `Npp16s` \*const pDst[3], int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*3 channel 16-bit signed planar color twist.*

- `NppStatus nppiColorTwist32f_16s_IP3R` (`Npp16s` \*const pSrcDst[3], int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*3 channel 16-bit signed planar in place color twist.*

- `NppStatus nppiColorTwist_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*3 channel 32-bit floating point color twist.*

- `NppStatus nppiColorTwist_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*3 channel 32-bit floating point in place color twist.*

- `NppStatus nppiColorTwist_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*4 channel 32-bit floating point color twist, with alpha copy.*

- `NppStatus nppiColorTwist_32f_C4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*4 channel 32-bit floating point in place color twist, not affecting Alpha.*

- `NppStatus nppiColorTwist_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*4 channel 32-bit floating point color twist, not affecting Alpha.*

- `NppStatus nppiColorTwist_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])

*4 channel 32-bit floating point in place color twist, not affecting Alpha.*

- `NppStatus nppiColorTwist_32f_P3R` (const `Npp32f` \*const pSrc[3], int nSrcStep, `Npp32f` \*const pDst[3], int nDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*3 channel 32-bit floating point planar color twist.*
- `NppStatus nppiColorTwist_32f_IP3R` (`Npp32f` \*const pSrcDst[3], int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` aTwist[3][4])  
*3 channel 32-bit floating point planar in place color twist.*

## ColorLUT

Perform image color processing using members of various types of color look up tables.

- `NppStatus nppiLUT_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)  
*8-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)  
*8-bit unsigned look-up-table in place color conversion.*
- `NppStatus nppiLUT_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*3 channel 8-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*3 channel 8-bit unsigned look-up-table in place color conversion.*
- `NppStatus nppiLUT_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])  
*4 channel 8-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_8u_C4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])  
*4 channel 8-bit unsigned look-up-table in place color conversion.*
- `NppStatus nppiLUT_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*4 channel 8-bit unsigned look-up-table in place color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)  
*16-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)

*16-bit unsigned look-up-table in place color conversion.*

- `NppStatus nppiLUT_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*3 channel 16-bit unsigned look-up-table color conversion.*

- `NppStatus nppiLUT_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*3 channel 16-bit unsigned look-up-table in place color conversion.*

- `NppStatus nppiLUT_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])

*4 channel 16-bit unsigned look-up-table color conversion.*

- `NppStatus nppiLUT_16u_C4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])

*4 channel 16-bit unsigned look-up-table in place color conversion.*

- `NppStatus nppiLUT_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)

*16-bit signed look-up-table color conversion.*

- `NppStatus nppiLUT_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)

*16-bit signed look-up-table in place color conversion.*

- `NppStatus nppiLUT_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*3 channel 16-bit signed look-up-table color conversion.*

- `NppStatus nppiLUT_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*3 channel 16-bit signed look-up-table in place color conversion.*

- `NppStatus nppiLUT_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])

*4 channel 16-bit signed look-up-table color conversion.*

- `NppStatus nppiLUT_16s_C4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])

*4 channel 16-bit signed look-up-table in place color conversion.*

- **NppStatus** **nppiLUT\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[3], const **Npp32s** \*pLevels[3], int nLevels[3])  
*4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.*
- **NppStatus** **nppiLUT\_16s\_AC4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[3], const **Npp32s** \*pLevels[3], int nLevels[3])  
*4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.*
- **NppStatus** **nppiLUT\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues, const **Npp32f** \*pLevels, int nLevels)  
*32-bit floating point look-up-table color conversion.*
- **NppStatus** **nppiLUT\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues, const **Npp32f** \*pLevels, int nLevels)  
*32-bit floating point look-up-table in place color conversion.*
- **NppStatus** **nppiLUT\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[3], const **Npp32f** \*pLevels[3], int nLevels[3])  
*3 channel 32-bit floating point look-up-table color conversion.*
- **NppStatus** **nppiLUT\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[3], const **Npp32f** \*pLevels[3], int nLevels[3])  
*3 channel 32-bit floating point look-up-table in place color conversion.*
- **NppStatus** **nppiLUT\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[4], const **Npp32f** \*pLevels[4], int nLevels[4])  
*4 channel 32-bit floating point look-up-table color conversion.*
- **NppStatus** **nppiLUT\_32f\_C4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[4], const **Npp32f** \*pLevels[4], int nLevels[4])  
*4 channel 32-bit floating point look-up-table in place color conversion.*
- **NppStatus** **nppiLUT\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[3], const **Npp32f** \*pLevels[3], int nLevels[3])  
*4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.*
- **NppStatus** **nppiLUT\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[3], const **Npp32f** \*pLevels[3], int nLevels[3])  
*4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.*

## ColorLUT\_Linear

Perform image color processing using linear interpolation between members of various types of color look up tables.

- **NppStatus** **nppiLUT\_Linear\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues, const **Npp32s** \*pLevels, int nLevels)  
*8-bit unsigned linear interpolated look-up-table color conversion.*



- `NppStatus nppiLUT_Linear_8u_C1IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)  
*8-bit unsigned linear interpolated look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_8u_C3R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)  
*3 channel 8-bit unsigned linear interpolated look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_8u_C3IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)  
*3 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_8u_C4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)  
*4 channel 8-bit unsigned linear interpolated look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_8u_C4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)  
*4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_8u_AC4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)  
*4 channel 8-bit unsigned linear interpolated look-up-table color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_Linear_8u_AC4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)  
*4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_Linear_16u_C1R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)  
*16-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_16u_C1IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)  
*16-bit unsigned look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_16u_C3R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)  
*3 channel 16-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_16u_C3IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)  
*3 channel 16-bit unsigned look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_16u_C4R` (`const Npp16u *pSrc`, `int nSrcStep`, `Npp16u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)  
*4 channel 16-bit unsigned look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_16u_C4IR` (`Npp16u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)  
*4 channel 16-bit unsigned look-up-table in place color conversion.*

- `NppStatus nppiLUT_Linear_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_Linear_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_Linear_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)  
*16-bit signed look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)  
*16-bit signed look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*3 channel 16-bit signed look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*3 channel 16-bit signed look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])  
*4 channel 16-bit signed look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_16s_C4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])  
*4 channel 16-bit signed look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_Linear_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])  
*4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.*
- `NppStatus nppiLUT_Linear_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pValues, const `Npp32f` \*pLevels, int nLevels)  
*32-bit floating point look-up-table color conversion.*
- `NppStatus nppiLUT_Linear_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pValues, const `Npp32f` \*pLevels, int nLevels)  
*32-bit floating point look-up-table in place color conversion.*
- `NppStatus nppiLUT_Linear_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pValues[3], const `Npp32f` \*pLevels[3], int nLevels[3])

*3 channel 32-bit floating point look-up-table color conversion.*

- `NppStatus nppiLUT_Linear_32f_C3IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)

*3 channel 32-bit floating point look-up-table in place color conversion.*

- `NppStatus nppiLUT_Linear_32f_C4R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[4]`, `const Npp32f *pLevels[4]`, `int nLevels[4]`)

*4 channel 32-bit floating point look-up-table color conversion.*

- `NppStatus nppiLUT_Linear_32f_C4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[4]`, `const Npp32f *pLevels[4]`, `int nLevels[4]`)

*4 channel 32-bit floating point look-up-table in place color conversion.*

- `NppStatus nppiLUT_Linear_32f_AC4R` (`const Npp32f *pSrc`, `int nSrcStep`, `Npp32f *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)

*4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_Linear_32f_AC4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)

*4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.*

## ColorLUT\_Cubic

Perform image color processing using linear interpolation between members of various types of color look up tables.

- `NppStatus nppiLUT_Cubic_8u_C1R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)

*8-bit unsigned cubic interpolated look-up-table color conversion.*

- `NppStatus nppiLUT_Cubic_8u_C1IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues`, `const Npp32s *pLevels`, `int nLevels`)

*8-bit unsigned cubic interpolated look-up-table in place color conversion.*

- `NppStatus nppiLUT_Cubic_8u_C3R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)

*3 channel 8-bit unsigned cubic interpolated look-up-table color conversion.*

- `NppStatus nppiLUT_Cubic_8u_C3IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[3]`, `const Npp32s *pLevels[3]`, `int nLevels[3]`)

*3 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.*

- `NppStatus nppiLUT_Cubic_8u_C4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)

*4 channel 8-bit unsigned cubic interpolated look-up-table color conversion.*

- `NppStatus nppiLUT_Cubic_8u_C4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32s *pValues[4]`, `const Npp32s *pLevels[4]`, `int nLevels[4]`)

*4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.*

- `NppStatus nppiLUT_Cubic_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*4 channel 8-bit unsigned cubic interpolated look-up-table color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_Cubic_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_Cubic_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)

*16-bit unsigned look-up-table color conversion.*

- `NppStatus nppiLUT_Cubic_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)

*16-bit unsigned look-up-table in place color conversion.*

- `NppStatus nppiLUT_Cubic_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*3 channel 16-bit unsigned look-up-table color conversion.*

- `NppStatus nppiLUT_Cubic_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*3 channel 16-bit unsigned look-up-table in place color conversion.*

- `NppStatus nppiLUT_Cubic_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])

*4 channel 16-bit unsigned look-up-table color conversion.*

- `NppStatus nppiLUT_Cubic_16u_C4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[4], const `Npp32s` \*pLevels[4], int nLevels[4])

*4 channel 16-bit unsigned look-up-table in place color conversion.*

- `NppStatus nppiLUT_Cubic_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_Cubic_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues[3], const `Npp32s` \*pLevels[3], int nLevels[3])

*4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_Cubic_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)

*16-bit signed look-up-table color conversion.*

- `NppStatus nppiLUT_Cubic_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pValues, const `Npp32s` \*pLevels, int nLevels)

*16-bit signed look-up-table in place color conversion.*

- **NppStatus nppiLUT\_Cubic\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[3], const **Npp32s** \*pLevels[3], int nLevels[3])  
*3 channel 16-bit signed look-up-table color conversion.*
- **NppStatus nppiLUT\_Cubic\_16s\_C3IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[3], const **Npp32s** \*pLevels[3], int nLevels[3])  
*3 channel 16-bit signed look-up-table in place color conversion.*
- **NppStatus nppiLUT\_Cubic\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[4], const **Npp32s** \*pLevels[4], int nLevels[4])  
*4 channel 16-bit signed look-up-table color conversion.*
- **NppStatus nppiLUT\_Cubic\_16s\_C4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[4], const **Npp32s** \*pLevels[4], int nLevels[4])  
*4 channel 16-bit signed look-up-table in place color conversion.*
- **NppStatus nppiLUT\_Cubic\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[3], const **Npp32s** \*pLevels[3], int nLevels[3])  
*4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.*
- **NppStatus nppiLUT\_Cubic\_16s\_AC4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32s** \*pValues[3], const **Npp32s** \*pLevels[3], int nLevels[3])  
*4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.*
- **NppStatus nppiLUT\_Cubic\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues, const **Npp32f** \*pLevels, int nLevels)  
*32-bit floating point look-up-table color conversion.*
- **NppStatus nppiLUT\_Cubic\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues, const **Npp32f** \*pLevels, int nLevels)  
*32-bit floating point look-up-table in place color conversion.*
- **NppStatus nppiLUT\_Cubic\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[3], const **Npp32f** \*pLevels[3], int nLevels[3])  
*3 channel 32-bit floating point look-up-table color conversion.*
- **NppStatus nppiLUT\_Cubic\_32f\_C3IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[3], const **Npp32f** \*pLevels[3], int nLevels[3])  
*3 channel 32-bit floating point look-up-table in place color conversion.*
- **NppStatus nppiLUT\_Cubic\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[4], const **Npp32f** \*pLevels[4], int nLevels[4])  
*4 channel 32-bit floating point look-up-table color conversion.*
- **NppStatus nppiLUT\_Cubic\_32f\_C4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[4], const **Npp32f** \*pLevels[4], int nLevels[4])  
*4 channel 32-bit floating point look-up-table in place color conversion.*
- **NppStatus nppiLUT\_Cubic\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** \*pValues[3], const **Npp32f** \*pLevels[3], int nLevels[3])  
*4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.*

- `NppStatus nppiLUT_Cubic_32f_AC4IR` (`Npp32f *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `const Npp32f *pValues[3]`, `const Npp32f *pLevels[3]`, `int nLevels[3]`)

*4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.*

## ColorLUT\_Trilinear

Perform image color processing using 3D trilinear interpolation between members of various types of color look up tables.

- `NppStatus nppiLUT_Trilinear_8u_C4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `Npp32u *pValues`, `Npp8u *pLevels[3]`, `int aLevels[3]`)

*Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, with alpha copy.*

- `NppStatus nppiLUT_Trilinear_8u_AC4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `Npp32u *pValues`, `Npp8u *pLevels[3]`, `int aLevels[3]`)

*Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, not affecting alpha.*

- `NppStatus nppiLUT_Trilinear_8u_AC4IR` (`Npp8u *pSrcDst`, `int nSrcDstStep`, `NppiSize oSizeROI`, `Npp32u *pValues`, `Npp8u *pLevels[3]`, `int aLevels[3]`)

*Four channel 8-bit unsigned 3D trilinear interpolated look-up-table in place color conversion, not affecting alpha.*

## ColorLUTPalette

Perform image color processing using various types of bit range restricted palette color look up tables.

- `NppStatus nppiLUTPalette_8u_C1R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u *pTable`, `int nBitSize`)

*One channel 8-bit unsigned bit range restricted palette look-up-table color conversion.*

- `NppStatus nppiLUTPalette_8u24u_C1R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u *pTable`, `int nBitSize`)

*One channel 8-bit unsigned bit range restricted 24-bit palette look-up-table color conversion with 24-bit destination output per pixel.*

- `NppStatus nppiLUTPalette_8u32u_C1R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp32u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp32u *pTable`, `int nBitSize`)

*One channel 8-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit destination output per pixel.*

- `NppStatus nppiLUTPalette_8u_C3R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u *pTables[3]`, `int nBitSize`)

*Three channel 8-bit unsigned bit range restricted palette look-up-table color conversion.*

- `NppStatus nppiLUTPalette_8u_C4R` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `NppiSize oSizeROI`, `const Npp8u *pTables[4]`, `int nBitSize`)

*Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion.*



- **NppStatus** **nppiLUTPalette\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pTables[3], int nBitSize)  
*Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.*
- **NppStatus** **nppiLUTPalette\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** \*pTable, int nBitSize)  
*One channel 16-bit unsigned bit range restricted palette look-up-table color conversion.*
- **NppStatus** **nppiLUTPalette\_16u8u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pTable, int nBitSize)  
*One channel 16-bit unsigned bit range restricted 8-bit unsigned palette look-up-table color conversion with 8-bit unsigned destination output per pixel.*
- **NppStatus** **nppiLUTPalette\_16u24u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pTable, int nBitSize)  
*One channel 16-bit unsigned bit range restricted 24-bit unsigned palette look-up-table color conversion with 24-bit unsigned destination output per pixel.*
- **NppStatus** **nppiLUTPalette\_16u32u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp32u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32u** \*pTable, int nBitSize)  
*One channel 16-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit unsigned destination output per pixel.*
- **NppStatus** **nppiLUTPalette\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** \*pTables[3], int nBitSize)  
*Three channel 16-bit unsigned bit range restricted palette look-up-table color conversion.*
- **NppStatus** **nppiLUTPalette\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** \*pTables[4], int nBitSize)  
*Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion.*
- **NppStatus** **nppiLUTPalette\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** \*pTables[3], int nBitSize)  
*Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.*
- **NppStatus** **nppiLUTPaletteSwap\_8u\_C3A0C4R** (const **Npp8u** \*pSrc, int nSrcStep, int nAlphaValue, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pTables[3], int nBitSize)  
*Three channel 8-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 8-bit unsigned destination output with alpha.*
- **NppStatus** **nppiLUTPaletteSwap\_16u\_C3A0C4R** (const **Npp16u** \*pSrc, int nSrcStep, int nAlphaValue, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** \*pTables[3], int nBitSize)  
*Three channel 16-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 16-bit unsigned destination output with alpha.*

### 7.48.1 Detailed Description

Routines for performing image color manipulation.

## 7.48.2 Function Documentation

### 7.48.2.1 **NppStatus nppiColorTwist32f\_16s\_AC4IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 16-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

#### Parameters:

- pSrcDst* in place packed pixel format image pointer.
- nSrcDstStep* in place packed pixel format image line step.
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- aTwist* The color twist matrix with floating-point coefficient values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.2 **NppStatus nppiColorTwist32f\_16s\_AC4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 16-bit signed color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

#### Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- aTwist* The color twist matrix with floating-point coefficient values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.3 **NppStatus nppiColorTwist32f\_16s\_C3IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

3 channel 16-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

#### Parameters:

- pSrcDst* in place packed pixel format image pointer.



*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.4 NppStatus nppiColorTwist32f\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])**

3 channel 16-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.5 NppStatus nppiColorTwist32f\_16s\_IP3R (Npp16s \*const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])**

3 channel 16-bit signed planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrcDst* in place planar pixel format image pointer array, one pointer per plane.

*nSrcDstStep* in place planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.6 **NppStatus nppiColorTwist32f\_16s\_P3R** (const Npp16s \*const *pSrc*[3], int *nSrcStep*, Npp16s \*const *pDst*[3], int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

3 channel 16-bit signed planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aTwist* The color twist matrix with floating-point coefficient values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.7 **NppStatus nppiColorTwist32f\_16u\_AC4IR** (Npp16u \**pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 16-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* Region-of-Interest (ROI).

*aTwist* The color twist matrix with floating-point coefficient values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.8 **NppStatus nppiColorTwist32f\_16u\_AC4R** (const Npp16u \**pSrc*, int *nSrcStep*, Npp16u \**pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

4 channel 16-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.48.2.9 NppStatus nppiColorTwist32f\_16u\_C3IR (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

3 channel 16-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* Region-of-Interest (ROI).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.48.2.10 NppStatus nppiColorTwist32f\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *aTwist*[3][4])

3 channel 16-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.48.2.11 NppStatus nppiColorTwist32f\_16u\_IP3R (Npp16u \*const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 16-bit unsigned planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

##### Parameters:

*pSrcDst* in place planar pixel format image pointer array, one pointer per plane.

*nSrcDstStep* in place planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.12 NppStatus nppiColorTwist32f\_16u\_P3R (const Npp16u \*const pSrc[3], int nSrcStep, Npp16u \*const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 16-bit unsigned planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.13 NppStatus nppiColorTwist32f\_8s\_AC4IR (Npp8s \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 8-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.14** `NppStatus nppiColorTwist32f_8s_AC4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit signed color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.15** `NppStatus nppiColorTwist32f_8s_C3IR (Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 8-bit signed in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.16** `NppStatus nppiColorTwist32f_8s_C3R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 8-bit signed color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.17** `NppStatus nppiColorTwist32f_8s_C4IR (Npp8s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit signed in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is unmodified.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.18** `NppStatus nppiColorTwist32f_8s_C4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit signed color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.19 NppStatus nppiColorTwist32f\_8s\_IP3R (Npp8s \*const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit signed planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

#### Parameters:

*pSrcDst* in place planar pixel format image pointer array, one pointer per plane.

*nSrcDstStep* in place planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.20 NppStatus nppiColorTwist32f\_8s\_P3R (const Npp8s \*const pSrc[3], int nSrcStep, Npp8s \*const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit signed planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.21 NppStatus nppiColorTwist32f\_8u\_AC4IR (Npp8u \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 8-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

#### Parameters:

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.22** `NppStatus nppiColorTwist32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit unsigned color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.23** `NppStatus nppiColorTwist32f_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 8-bit unsigned in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.24** `NppStatus nppiColorTwist32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 8-bit unsigned color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.



**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.25** `NppStatus nppiColorTwist32f_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit unsigned in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is unmodified.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.26** `NppStatus nppiColorTwist32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 8-bit unsigned color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.27 NppStatus nppiColorTwist32f\_8u\_IP3R (Npp8u \*const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit unsigned planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

##### Parameters:

*pSrcDst* in place planar pixel format image pointer array, one pointer per plane.

*nSrcDstStep* in place planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.28 NppStatus nppiColorTwist32f\_8u\_P3R (const Npp8u \*const pSrc[3], int nSrcStep, Npp8u \*const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 8-bit unsigned planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.48.2.29 NppStatus nppiColorTwist\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

4 channel 32-bit floating point in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.30** `NppStatus nppiColorTwist_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.31** `NppStatus nppiColorTwist_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.

*nSrcDstStep* in place packed pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.32** `NppStatus nppiColorTwist_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

3 channel 32-bit floating point color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.33** `NppStatus nppiColorTwist_32f_C4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point in place color twist, not affecting Alpha.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is not modified.

**Parameters:**

*pSrcDst* in place packed pixel format image pointer.  
*nSrcDstStep* in place packed pixel format image line step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.48.2.34** `NppStatus nppiColorTwist_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])`

4 channel 32-bit floating point color twist, with alpha copy.

An input color twist matrix with floating-point coefficient values is applied with in ROI. Alpha channel is the last channel and is copied unmodified from the source pixel to the destination pixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aTwist* The color twist matrix with floating-point coefficient values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.35 NppStatus nppiColorTwist\_32f\_IP3R (Npp32f \*const pSrcDst[3], int nSrcDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 32-bit floating point planar in place color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

#### Parameters:

*pSrcDst* in place planar pixel format image pointer array, one pointer per plane.

*nSrcDstStep* in place planar pixel format image line step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.36 NppStatus nppiColorTwist\_32f\_P3R (const Npp32f \*const pSrc[3], int nSrcStep, Npp32f \*const pDst[3], int nDstStep, NppiSize oSizeROI, const Npp32f aTwist[3][4])

3 channel 32-bit floating point planar color twist.

An input color twist matrix with floating-point coefficient values is applied within ROI.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aTwist* The color twist matrix with floating-point coefficient values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.48.2.37 NppStatus nppiLUT\_16s\_AC4IR (Npp16s \*pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s \*pValues[3], const Npp32s \*pLevels[3], int nLevels[3])

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.38** `NppStatus nppiLUT_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.39** `NppStatus nppiLUT_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)  
*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)  
*nLevels* Number of user defined number of input/output mapping points (levels)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.40** `NppStatus nppiLUT_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)  
*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)  
*nLevels* Number of user defined number of input/output mapping points (levels)

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.41** `NppStatus nppiLUT_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.42** `NppStatus nppiLUT_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.43** `NppStatus nppiLUT_16s_C4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.



**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.44** `NppStatus nppiLUT_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.45 `NppStatus nppiLUT_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.46 `NppStatus nppiLUT_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- `NPP_LUT_NUMBER_OF_LEVELS_ERROR` if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.47 NppStatus nppiLUT\_16u\_C1IR (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*, const Npp32s \* *pLevels*, int *nLevels*)

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.48 NppStatus nppiLUT\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*, const Npp32s \* *pLevels*, int *nLevels*)

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.49 **NppStatus nppiLUT\_16u\_C3IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.50 **NppStatus nppiLUT\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.51 NppStatus nppiLUT\_16u\_C4IR (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.52 NppStatus nppiLUT\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.53 **NppStatus nppiLUT\_32f\_AC4IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.54 **NppStatus nppiLUT\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.55 `NppStatus nppiLUT_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.56 `NppStatus nppiLUT_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.57 **NppStatus nppiLUT\_32f\_C3IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.58 **NppStatus nppiLUT\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).



### 7.48.2.59 NppStatus nppiLUT\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.60 NppStatus nppiLUT\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.61 **NppStatus nppiLUT\_8u\_AC4IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.62 **NppStatus nppiLUT\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.63 `NppStatus nppiLUT_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.64 `NppStatus nppiLUT_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

**7.48.2.65** `NppStatus nppiLUT_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

**7.48.2.66** `NppStatus nppiLUT_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.67 `NppStatus nppiLUT_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.68 `NppStatus nppiLUT_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points with no interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.69 NppStatus nppiLUT\_Cubic\_16s\_AC4IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.70 NppStatus nppiLUT\_Cubic\_16s\_AC4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.71 `NppStatus nppiLUT_Cubic_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.72 `NppStatus nppiLUT_Cubic_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.73 **NppStatus nppiLUT\_Cubic\_16s\_C3IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.74 **NppStatus nppiLUT\_Cubic\_16s\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).



### 7.48.2.75 `NppStatus nppiLUT_Cubic_16s_C4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.76 `NppStatus nppiLUT_Cubic_16s_C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.77 **NppStatus nppiLUT\_Cubic\_16u\_AC4IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.78 **NppStatus nppiLUT\_Cubic\_16u\_AC4R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.79 NppStatus nppiLUT\_Cubic\_16u\_C1IR (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*, const Npp32s \* *pLevels*, int *nLevels*)

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.80 NppStatus nppiLUT\_Cubic\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*, const Npp32s \* *pLevels*, int *nLevels*)

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.81 `NppStatus nppiLUT_Cubic_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.82 `NppStatus nppiLUT_Cubic_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.83 NppStatus nppiLUT\_Cubic\_16u\_C4IR (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.84 NppStatus nppiLUT\_Cubic\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.85 **NppStatus nppiLUT\_Cubic\_32f\_AC4IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.86 **NppStatus nppiLUT\_Cubic\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.87 `NppStatus nppiLUT_Cubic_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.88 `NppStatus nppiLUT_Cubic_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pValues, const Npp32f * pLevels, int nLevels)`

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.89 **NppStatus nppiLUT\_Cubic\_32f\_C3IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.90 **NppStatus nppiLUT\_Cubic\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).



### 7.48.2.91 NppStatus nppiLUT\_Cubic\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.92 NppStatus nppiLUT\_Cubic\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.93 **NppStatus nppiLUT\_Cubic\_8u\_AC4IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through cubic interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.94 **NppStatus nppiLUT\_Cubic\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through cubic interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.95 NppStatus nppiLUT\_Cubic\_8u\_C1R (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*, const Npp32s \* *pLevels*, int *nLevels*)

8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.96 NppStatus nppiLUT\_Cubic\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*, const Npp32s \* *pLevels*, int *nLevels*)

8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.97 `NppStatus nppiLUT_Cubic_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.98 `NppStatus nppiLUT_Cubic_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

3 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.99 `NppStatus nppiLUT_Cubic_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned cubic interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

### 7.48.2.100 `NppStatus nppiLUT_Cubic_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned cubic interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through cubic interpolation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

**7.48.2.101** `NppStatus nppiLUT_Linear_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit signed look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.102** `NppStatus nppiLUT_Linear_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit signed look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.103** `NppStatus nppiLUT_Linear_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.104** `NppStatus nppiLUT_Linear_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.105 **NppStatus nppiLUT\_Linear\_16s\_C3IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.106 **NppStatus nppiLUT\_Linear\_16s\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).



#### 7.48.2.107 NppStatus nppiLUT\_Linear\_16s\_C4IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit signed look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.108 NppStatus nppiLUT\_Linear\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit signed look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.109** `NppStatus nppiLUT_Linear_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.110** `NppStatus nppiLUT_Linear_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[3], const Npp32s * pLevels[3], int nLevels[3])`

4 channel 16-bit unsigned look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.111** `NppStatus nppiLUT_Linear_16u_C11R (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

**7.48.2.112** `NppStatus nppiLUT_Linear_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.113 **NppStatus nppiLUT\_Linear\_16u\_C3IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.114 **NppStatus nppiLUT\_Linear\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

3 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.115 **NppStatus nppiLUT\_Linear\_16u\_C4IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.116 **NppStatus nppiLUT\_Linear\_16u\_C4R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4])

4 channel 16-bit unsigned look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.117 **NppStatus nppiLUT\_Linear\_32f\_AC4IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.118 **NppStatus nppiLUT\_Linear\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

4 channel 32-bit floating point look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points using no interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.119 **NppStatus nppiLUT\_Linear\_32f\_C1IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*, const Npp32f \* *pLevels*, int *nLevels*)

32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.120 **NppStatus nppiLUT\_Linear\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*, const Npp32f \* *pLevels*, int *nLevels*)

32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.121 **NppStatus nppiLUT\_Linear\_32f\_C3IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

3 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.122 **NppStatus nppiLUT\_Linear\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3])

3 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).



### 7.48.2.123 **NppStatus nppiLUT\_Linear\_32f\_C4IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points using no interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

### 7.48.2.124 **NppStatus nppiLUT\_Linear\_32f\_C4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pValues*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4])

4 channel 32-bit floating point look-up-table color conversion.

The LUT is derived from a set of user defined mapping points using linear interpolation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 1024 (the current size limit).

#### 7.48.2.125 **NppStatus nppiLUT\_Linear\_8u\_AC4IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.126 **NppStatus nppiLUT\_Linear\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32s \* *pValues*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3])

4 channel 8-bit unsigned linear interpolated look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points through linear interpolation. Alpha channel is the last channel and is not processed.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the *pValues* and *pLevels* pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

**7.48.2.127** `NppStatus nppiLUT_Linear_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

**7.48.2.128** `NppStatus nppiLUT_Linear_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues, const Npp32s * pLevels, int nLevels)`

8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be device memory pointers.

>>>>>> <<<<<<<

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Pointer to an array of user defined OUTPUT values (this is now a device memory pointer)

*pLevels* Pointer to an array of user defined INPUT values (this is now a device memory pointer)

*nLevels* Number of user defined number of input/output mapping points (levels)

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.129 NppStatus nppiLUT\_Linear\_8u\_C3IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s \* pValues[3], const Npp32s \* pLevels[3], int nLevels[3])

3 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.130 NppStatus nppiLUT\_Linear\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32s \* pValues[3], const Npp32s \* pLevels[3], int nLevels[3])

3 channel 8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the pValues and pLevels pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR** if the number of levels is less than 2 or greater than 256.

**7.48.2.131** `NppStatus nppiLUT_Linear_8u_C4IR(Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned linear interpolated look-up-table in place color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR** if the number of levels is less than 2 or greater than 256.

**7.48.2.132** `NppStatus nppiLUT_Linear_8u_C4R(const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32s * pValues[4], const Npp32s * pLevels[4], int nLevels[4])`

4 channel 8-bit unsigned linear interpolated look-up-table color conversion.

The LUT is derived from a set of user defined mapping points through linear interpolation.

>>>>>> ATTENTION ATTENTION <<<<<<<

NOTE: As of the 5.0 release of NPP, the *pValues* and *pLevels* pointers need to be host memory pointers to arrays of device memory pointers.

>>>>>> <<<<<<<

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT values.

*pLevels* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined INPUT values.

*nLevels* Host pointer to an array of 4 user defined number of input/output mapping points, one per color CHANNEL.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

#### 7.48.2.133 NppStatus nppiLUT\_Trilinear\_8u\_AC4IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, Npp32u \* *pValues*, Npp8u \* *pLevels*[3], int *aLevels*[3])

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table in place color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Device pointer *aLevels*[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x \* y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.

*pLevels* Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.

*aLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. *aLevels*[0] represents the number of x axis levels (Red), *aLevels*[1] represents the number of y axis levels (Green), and *aLevels*[2] represents the number of z axis levels (Blue).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

**7.48.2.134 NppStatus nppiLUT\_Trilinear\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32u \* *pValues*, Npp8u \* *pLevels*[3], int *aLevels*[3])**

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, not affecting alpha.

Alpha channel is the last channel and is not processed.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pValues* Device pointer to *aLevels*[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x \* y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.

*pLevels* Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.

*aLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. *aLevels*[0] represents the number of x axis levels (Red), *aLevels*[1] represents the number of y axis levels (Green), and *aLevels*[2] represents the number of z axis levels (Blue).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_NUMBER\\_OF\\_LEVELS\\_ERROR](#) if the number of levels is less than 2 or greater than 256.

**7.48.2.135 NppStatus nppiLUT\_Trilinear\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32u \* *pValues*, Npp8u \* *pLevels*[3], int *aLevels*[3])**

Four channel 8-bit unsigned 3D trilinear interpolated look-up-table color conversion, with alpha copy.

Alpha channel is the last channel and is copied to the destination unmodified.

The LUT is derived from a set of user defined mapping points through trilinear interpolation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pValues* Device pointer to aLevels[2] number of contiguous 2D x,y planes of 4-byte packed RGBX values containing the user defined base OUTPUT values at that x,y, and z (R,G,B) level location. Each level must contain x \* y 4-byte packed pixel values (4th byte is used for alignment only and is ignored) in row (x) order.

*pLevels* Host pointer to an array of 3 host pointers, one per cube edge, pointing to user defined INPUT level values.

*aLevels* Host pointer to an array of 3 user defined number of input/output mapping points, one per 3D cube edge. aLevels[0] represents the number of x axis levels (Red), aLevels[1] represents the number of y axis levels (Green), and aLevels[2] represents the number of z axis levels (Blue).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR** if the number of levels is less than 2 or greater than 256.

#### 7.48.2.136 NppStatus nppiLUTPalette\_16u24u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pTable, int nBitSize)

One channel 16-bit unsigned bit range restricted 24-bit unsigned palette look-up-table color conversion with 24-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step (3 unsigned bytes per pixel).

*oSizeROI* Region-of-Interest (ROI).

*pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_PALETTE\_BITSIZE\_ERROR** if nBitSize is < 1 or > 16.



### 7.48.2.137 **NppStatus nppiLUTPalette\_16u32u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32u \* *pTable*, int *nBitSize*)

One channel 16-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#) (4 bytes per pixel).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 16.

### 7.48.2.138 **NppStatus nppiLUTPalette\_16u8u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pTable*, int *nBitSize*)

One channel 16-bit unsigned bit range restricted 8-bit unsigned palette look-up-table color conversion with 8-bit unsigned destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#) (1 unsigned byte per pixel).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 16.

#### 7.48.2.139 **NppStatus nppiLUTPalette\_16u\_AC4R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u \* *pTables*[3], int *nBitSize*)

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 16.

#### 7.48.2.140 **NppStatus nppiLUTPalette\_16u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u \* *pTable*, int *nBitSize*)

One channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 16.

#### 7.48.2.141 **NppStatus nppiLUTPalette\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u \* *pTables*[3], int *nBitSize*)

Three channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_PALETTE\_BITSIZE\_ERROR** if *nBitSize* is < 1 or > 16.

#### 7.48.2.142 **NppStatus nppiLUTPalette\_16u\_C4R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u \* *pTables*[4], int *nBitSize*)

Four channel 16-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTables* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_PALETTE\_BITSIZE\_ERROR** if *nBitSize* is < 1 or > 16.

#### 7.48.2.143 NppStatus nppiLUTPalette\_8u24u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pTable*, int *nBitSize*)

One channel 8-bit unsigned bit range restricted 24-bit palette look-up-table color conversion with 24-bit destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step (3 bytes per pixel).

*oSizeROI* Region-of-Interest (ROI).

*pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

*nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 8.

#### 7.48.2.144 NppStatus nppiLUTPalette\_8u32u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32u \* *pTable*, int *nBitSize*)

One channel 8-bit unsigned bit range restricted 32-bit palette look-up-table color conversion with 32-bit destination output per pixel.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step (4 bytes per pixel).

*oSizeROI* Region-of-Interest (ROI).

*pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

*nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 8.

#### 7.48.2.145 NppStatus nppiLUTPalette\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pTables*[3], int *nBitSize*)

Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion, not affecting Alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. Alpha channel is the last channel and is not processed.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

*nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_PALETTE\_BITSIZE\_ERROR** if *nBitSize* is < 1 or > 8.

#### 7.48.2.146 NppStatus nppiLUTPalette\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pTable*, int *nBitSize*)

One channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTable* Pointer to an array of user defined OUTPUT palette values (this is a device memory pointer)

*nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- **NPP\_LUT\_PALETTE\_BITSIZE\_ERROR** if *nBitSize* is < 1 or > 8.

#### 7.48.2.147 NppStatus nppiLUTPalette\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pTables*[3], int *nBitSize*)

Three channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

*nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 8.

#### 7.48.2.148 NppStatus nppiLUTPalette\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pTables*[4], int *nBitSize*)

Four channel 8-bit unsigned bit range restricted palette look-up-table color conversion.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pTables* Host pointer to an array of 4 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values.

*nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 8.

#### 7.48.2.149 **NppStatus nppiLUTPaletteSwap\_16u\_C3A0C4R** (const Npp16u \* *pSrc*, int *nSrcStep*, int *nAlphaValue*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u \* *pTables*[3], int *nBitSize*)

Three channel 16-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 16-bit unsigned destination output with alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. This function also reverses the source pixel channel order in the destination so the Alpha channel is the first channel.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#) (3 unsigned short integers per pixel).

*nAlphaValue* Signed alpha value that will be used to initialize the pixel alpha channel position in all modified destination pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#) (4 unsigned short integers per pixel with alpha).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values. Alpha values < 0 or > 65535 will cause destination pixel alpha channel values to be unmodified.

*nBitSize* Number of least significant bits (must be > 0 and <= 16) of each source pixel value to use as index into palette table during conversion.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if *nBitSize* is < 1 or > 16.

#### 7.48.2.150 **NppStatus nppiLUTPaletteSwap\_8u\_C3A0C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, int *nAlphaValue*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pTables*[3], int *nBitSize*)

Three channel 8-bit unsigned source bit range restricted palette look-up-table color conversion to four channel 8-bit unsigned destination output with alpha.

The LUT is derived from a set of user defined mapping points in a palette and source pixels are then processed using a restricted bit range when looking up palette values. This function also reverses the source pixel channel order in the destination so the Alpha channel is the first channel.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#) (3 bytes per pixel).

*nAlphaValue* Signed alpha value that will be used to initialize the pixel alpha channel position in all modified destination pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#) (4 bytes per pixel with alpha).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pTables* Host pointer to an array of 3 device memory pointers, one per color CHANNEL, pointing to user defined OUTPUT palette values. Alpha values < 0 or > 255 will cause destination pixel alpha channel values to be unmodified.

*nBitSize* Number of least significant bits (must be > 0 and <= 8) of each source pixel value to use as index into palette table during conversion.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- [NPP\\_LUT\\_PALETTE\\_BITSIZE\\_ERROR](#) if nBitSize is < 1 or > 8.



## 7.49 Compression

Image compression primitives.

### Modules

- [Quantization Functions](#)

### Typedefs

- typedef struct [NppiDecodeHuffmanSpec](#) [NppiDecodeHuffmanSpec](#)

### Functions

- [NppStatus nppiDecodeHuffmanSpecGetBufSize\\_JPEG](#) (int \*pSize)  
*Returns the length of the NppiDecodeHuffmanSpec structure.*
- [NppStatus nppiDecodeHuffmanSpecInitHost\\_JPEG](#) (const [Npp8u](#) \*pRawHuffmanTable, [NppiHuffmanTableType](#) eTableType, [NppiDecodeHuffmanSpec](#) \*pHuffmanSpec)  
*Creates a Huffman table in a format that is suitable for the decoder on the host.*
- [NppStatus nppiDecodeHuffmanSpecInitAllocHost\\_JPEG](#) (const [Npp8u](#) \*pRawHuffmanTable, [NppiHuffmanTableType](#) eTableType, [NppiDecodeHuffmanSpec](#) \*\*ppHuffmanSpec)  
*Allocates memory and creates a Huffman table in a format that is suitable for the decoder on the host.*
- [NppStatus nppiDecodeHuffmanSpecFreeHost\\_JPEG](#) ([NppiDecodeHuffmanSpec](#) \*pHuffmanSpec)  
*Frees the host memory allocated by nppiDecodeHuffmanSpecInitAllocHost\_JPEG.*
- [NppStatus nppiDecodeHuffmanScanHost\\_JPEG\\_8u16s\\_P1R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nLength, [Npp32s](#) restartInterval, [Npp32s](#) Ss, [Npp32s](#) Se, [Npp32s](#) Ah, [Npp32s](#) Al, [Npp16s](#) \*pDst, [Npp32s](#) nDstStep, [NppiDecodeHuffmanSpec](#) \*pHuffmanTableDC, [NppiDecodeHuffmanSpec](#) \*pHuffmanTableAC, [NppiSize](#) oSizeROI)  
*Huffman Decoding of the JPEG decoding on the host.*
- [NppStatus nppiDecodeHuffmanScanHost\\_JPEG\\_8u16s\\_P3R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nLength, [Npp32s](#) nRestartInterval, [Npp32s](#) nSs, [Npp32s](#) nSe, [Npp32s](#) nAh, [Npp32s](#) nAl, [Npp16s](#) \*apDst[3], [Npp32s](#) aDstStep[3], [NppiDecodeHuffmanSpec](#) \*apHuffmanDCTable[3], [NppiDecodeHuffmanSpec](#) \*apHuffmanACTable[3], [NppiSize](#) aSizeROI[3])  
*Huffman Decoding of the JPEG decoding on the host.*

### 7.49.1 Detailed Description

Image compression primitives.

The JPEG standard defines a flow of level shift, DCT and quantization for forward JPEG transform and inverse level shift, IDCT and de-quantization for inverse JPEG transform. This group has the functions for both forward and inverse functions.

## 7.49.2 Typedef Documentation

### 7.49.2.1 typedef struct NppiDecodeHuffmanSpec NppiDecodeHuffmanSpec

## 7.49.3 Function Documentation

### 7.49.3.1 NppStatus nppiDecodeHuffmanScanHost\_JPEG\_8u16s\_P1R (const Npp8u \* *pSrc*, Npp32s *nLength*, Npp32s *restartInterval*, Npp32s *Ss*, Npp32s *Se*, Npp32s *Ah*, Npp32s *Al*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiDecodeHuffmanSpec \* *pHuffmanTableDC*, NppiDecodeHuffmanSpec \* *pHuffmanTableAC*, NppiSize *oSizeROI*)

Huffman Decoding of the JPEG decoding on the host.

Input is expected in byte stuffed huffman encoded JPEG scan and output is expected to be 64x1 macro blocks.

#### Parameters:

*pSrc* Byte-stuffed huffman encoded JPEG scan.

*nLength* Byte length of the input.

*restartInterval* Restart Interval, see JPEG standard.

*Ss* Start Coefficient, see JPEG standard.

*Se* End Coefficient, see JPEG standard.

*Ah* Bit Approximation High, see JPEG standard.

*Al* Bit Approximation Low, see JPEG standard.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pHuffmanTableDC* DC Huffman table.

*pHuffmanTableAC* AC Huffman table.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) For negative input height/width or not a multiple of 8 width/height.
- [NPP\\_STEP\\_ERROR](#) If input image width is not multiple of 8 or does not match ROI.
- [NPP\\_NULL\\_POINTER\\_ERROR](#) If the destination pointer is 0.

### 7.49.3.2 NppStatus nppiDecodeHuffmanScanHost\_JPEG\_8u16s\_P3R (const Npp8u \* *pSrc*, Npp32s *nLength*, Npp32s *nRestartInterval*, Npp32s *nSs*, Npp32s *nSe*, Npp32s *nAh*, Npp32s *nAl*, Npp16s \* *apDst*[3], Npp32s *aDstStep*[3], NppiDecodeHuffmanSpec \* *apHuffmanDCTable*[3], NppiDecodeHuffmanSpec \* *apHuffmanACTable*[3], NppiSize *aSizeROI*[3])

Huffman Decoding of the JPEG decoding on the host.

Input is expected in byte stuffed huffman encoded JPEG scan and output is expected to be 64x1 macro blocks.

**Parameters:**

- pSrc* Byte-stuffed huffman encoded JPEG scan.
- nLength* Byte length of the input.
- nRestartInterval* Restart Interval, see JPEG standard.
- nSs* Start Coefficient, see JPEG standard.
- nSe* End Coefficient, see JPEG standard.
- nAh* Bit Approximation High, see JPEG standard.
- nAl* Bit Approximation Low, see JPEG standard.
- apDst* [Destination-Image Pointer](#).
- aDstStep* [Destination-Image Line Step](#).
- apHuffmanDCTable* DC Huffman tables.
- apHuffmanACTable* AC Huffman tables.
- aSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

Error codes:

- [NPP\\_SIZE\\_ERROR](#) For negative input height/width or not a multiple of 8 width/height.
- [NPP\\_STEP\\_ERROR](#) If input image width is not multiple of 8 or does not match ROI.
- [NPP\\_NULL\\_POINTER\\_ERROR](#) If the destination pointer is 0.

### 7.49.3.3 **NppStatus nppiDecodeHuffmanSpecFreeHost\_JPEG (NppiDecodeHuffmanSpec \* pHuffmanSpec)**

Frees the host memory allocated by nppiDecodeHuffmanSpecInitAllocHost\_JPEG.

**Parameters:**

- pHuffmanSpec* Pointer to the Huffman table for the decoder

### 7.49.3.4 **NppStatus nppiDecodeHuffmanSpecGetBufSize\_JPEG (int \* pSize)**

Returns the length of the NppiDecodeHuffmanSpec structure.

**Parameters:**

- pSize* Pointer to a variable that will receive the length of the NppiDecodeHuffmanSpec structure.

**Returns:**

Error codes:

- [NPP\\_NULL\\_POINTER\\_ERROR](#) If one of the pointers is 0.

**7.49.3.5 NppStatus nppiDecodeHuffmanSpecInitAllocHost\_JPEG (const Npp8u \*  
*pRawHuffmanTable*, NppiHuffmanTableType *eTableType*, NppiDecodeHuffmanSpec \*\*  
*ppHuffmanSpec*)**

Allocates memory and creates a Huffman table in a format that is suitable for the decoder on the host.

**Parameters:**

*pRawHuffmanTable* Huffman table formatted as specified in the JPEG standard.

*eTableType* Enum specifying type of table (nppiDCTable or nppiACTable).

*ppHuffmanSpec* Pointer to returned pointer to the Huffman table for the decoder

**Returns:**

Error codes:

- [NPP\\_NULL\\_POINTER\\_ERROR](#) If one of the pointers is 0.

**7.49.3.6 NppStatus nppiDecodeHuffmanSpecInitHost\_JPEG (const Npp8u \**pRawHuffmanTable*,  
NppiHuffmanTableType *eTableType*, NppiDecodeHuffmanSpec \**pHuffmanSpec*)**

Creates a Huffman table in a format that is suitable for the decoder on the host.

**Parameters:**

*pRawHuffmanTable* Huffman table formatted as specified in the JPEG standard.

*eTableType* Enum specifying type of table (nppiDCTable or nppiACTable).

*pHuffmanSpec* Pointer to the Huffman table for the decoder

**Returns:**

Error codes:

- [NPP\\_NULL\\_POINTER\\_ERROR](#) If one of the pointers is 0.

## 7.50 Quantization Functions

### Typedefs

- typedef struct [NppiDCTState](#) [NppiDCTState](#)

### Functions

- [NppStatus](#) [nppiQuantFwdRawTableInit\\_JPEG\\_8u](#) ([Npp8u](#) \*hpQuantRawTable, int nQualityFactor)

*Apply quality factor to raw 8-bit quantization table.*

- [NppStatus](#) [nppiQuantFwdTableInit\\_JPEG\\_8u16u](#) (const [Npp8u](#) \*hpQuantRawTable, [Npp16u](#) \*hpQuantFwdRawTable)

*Initializes a quantization table for [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#).*

- [NppStatus](#) [nppiQuantInvTableInit\\_JPEG\\_8u16u](#) (const [Npp8u](#) \*hpQuantRawTable, [Npp16u](#) \*hpQuantFwdRawTable)

*Initializes a quantization table for [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R\(\)](#).*

- [NppStatus](#) [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, const [Npp16u](#) \*pQuantFwdTable, [NppiSize](#) oSizeROI)

*Forward DCT, quantization and level shift part of the JPEG encoding.*

- [NppStatus](#) [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, const [Npp16u](#) \*pQuantInvTable, [NppiSize](#) oSizeROI)

*Inverse DCT, de-quantization and level shift part of the JPEG decoding.*

- [NppStatus](#) [nppiDCTInitAlloc](#) ([NppiDCTState](#) \*\*ppState)

*Initializes DCT state structure and allocates additional resources.*

- [NppStatus](#) [nppiDCTFree](#) ([NppiDCTState](#) \*pState)

*Frees the additional resources of the DCT state structure.*

- [NppStatus](#) [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\\_NEW](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, const [Npp8u](#) \*pQuantizationTable, [NppiSize](#) oSizeROI, [NppiDCTState](#) \*pState)

*Forward DCT, quantization and level shift part of the JPEG encoding.*

- [NppStatus](#) [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R\\_NEW](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, const [Npp8u](#) \*pQuantizationTable, [NppiSize](#) oSizeROI, [NppiDCTState](#) \*pState)

*Inverse DCT, de-quantization and level shift part of the JPEG decoding.*

## 7.50.1 Typedef Documentation

### 7.50.1.1 typedef struct NppiDCTState NppiDCTState

## 7.50.2 Function Documentation

### 7.50.2.1 NppStatus nppiDCTFree (NppiDCTState \* *pState*)

Frees the additional resources of the DCT state structure.

See also:

[nppiDCTInitAlloc](#)

**Parameters:**

*pState* Pointer to DCT state structure.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning  
 NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value  
 NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pState pointer is NULL

### 7.50.2.2 NppStatus nppiDCTInitAlloc (NppiDCTState \*\* *ppState*)

Initializes DCT state structure and allocates additional resources.

See also:

[nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\\_NEW\(\)](#), [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R\\_NEW](#).

**Parameters:**

*ppState* Pointer to pointer to DCT state structure.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning  
 NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value  
 NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

### 7.50.2.3 NppStatus nppiDCTQuantFwd8x8LS\_JPEG\_8u16s\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, const Npp16u \* *pQuantFwdTable*, NppiSize *oSizeROI*)

Forward DCT, quantization and level shift part of the JPEG encoding.

Input is expected in 8x8 macro blocks and output is expected to be in 64x1 macro blocks.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pQuantFwdTable* Forward quantization tables for JPEG encoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

#### 7.50.2.4 `NppStatus nppiDCTQuantFwd8x8LS_JPEG_8u16s_C1R_NEW` (`const Npp8u *pSrc`, `int nSrcStep`, `Npp16s *pDst`, `int nDstStep`, `const Npp8u *pQuantizationTable`, `NppiSize oSizeROI`, `NppiDCTState *pState`)

Forward DCT, quantization and level shift part of the JPEG encoding.

Input is expected in 8x8 macro blocks and output is expected to be in 64x1 macro blocks. The new version of the primitive takes the ROI in image pixel size and works with DCT coefficients that are in zig-zag order.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Image width in pixels x 8 x `sizeof(Npp16s)`.

*pQuantizationTable* Quantization Table in zig-zag order.

*oSizeROI* Region-of-Interest (ROI).

*pState* Pointer to DCT state structure. This structure must be initialized allocated and initialized using `nppiDCTInitAlloc()`.

#### Returns:

Error codes:

- `NPP_SIZE_ERROR` For negative input height/width or not a multiple of 8 width/height.
- `NPP_STEP_ERROR` If input image width is not multiple of 8 or does not match ROI.
- `NPP_NULL_POINTER_ERROR` If the destination pointer is 0.

#### 7.50.2.5 `NppStatus nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R` (`const Npp16s *pSrc`, `int nSrcStep`, `Npp8u *pDst`, `int nDstStep`, `const Npp16u *pQuantInvTable`, `NppiSize oSizeROI`)

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

Input is expected in 64x1 macro blocks and output is expected to be in 8x8 macro blocks.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Image width in pixels x 8 x sizeof(Npp16s).

*pDst* Destination-Image Pointer.

*nDstStep* Image width in pixels x 8 x sizeof(Npp16s).

*pQuantInvTable* Inverse quantization tables for JPEG decoding created using `nppiQuantInvTableInit_JPEG_8u16u()`.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Error codes:

- **NPP\_SIZE\_ERROR** For negative input height/width or not a multiple of 8 width/height.
- **NPP\_STEP\_ERROR** If input image width is not multiple of 8 or does not match ROI.
- **NPP\_NULL\_POINTER\_ERROR** If the destination pointer is 0.

**7.50.2.6** `NppStatus nppiDCTQuantInv8x8LS_JPEG_16s8u_C1R_NEW (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, const Npp8u * pQuantizationTable, NppiSize oSizeROI, NppiDCTState * pState)`

Inverse DCT, de-quantization and level shift part of the JPEG decoding.

Input is expected in 64x1 macro blocks and output is expected to be in 8x8 macro blocks. The new version of the primitive takes the ROI in image pixel size and works with DCT coefficients that are in zig-zag order.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Image width in pixels x 8 x sizeof(Npp16s).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pQuantizationTable* Quantization Table in zig-zag order.

*oSizeROI* Region-of-Interest (ROI).

*pState* Pointer to DCT state structure. This structure must be initialized allocated and initialized using `nppiDCTInitAlloc()`.

**Returns:**

Error codes:

- **NPP\_SIZE\_ERROR** For negative input height/width or not a multiple of 8 width/height.
- **NPP\_STEP\_ERROR** If input image width is not multiple of 8 or does not match ROI.
- **NPP\_NULL\_POINTER\_ERROR** If the destination pointer is 0.



### 7.50.2.7 NppStatus nppiQuantFwdRawTableInit\_JPEG\_8u (Npp8u \* *hpQuantRawTable*, int *nQualityFactor*)

Apply quality factor to raw 8-bit quantization table.

This is effectively an in-place method that modifies a given raw quantization table based on a quality factor. Note that this method is a host method and that the pointer to the raw quantization table is a host pointer.

#### Parameters:

*hpQuantRawTable* Raw quantization table.

*nQualityFactor* Quality factor for the table. Range is [1:100].

#### Returns:

Error code: [NPP\\_NULL\\_POINTER\\_ERROR](#) is returned if *hpQuantRawTable* is 0.

### 7.50.2.8 NppStatus nppiQuantFwdTableInit\_JPEG\_8u16u (const Npp8u \* *hpQuantRawTable*, Npp16u \* *hpQuantFwdRawTable*)

Initializes a quantization table for [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#).

The method creates a 16-bit version of the raw table and converts the data order from zigzag layout to original row-order layout since raw quantization tables are typically stored in zigzag format.

This method is a host method. It consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) function.

#### Parameters:

*hpQuantRawTable* Host pointer to raw quantization table as returned by [nppiQuantFwdRawTableInit\\_JPEG\\_8u\(\)](#). The raw quantization table is assumed to be in zigzag order.

*hpQuantFwdRawTable* Forward quantization table for use with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#).

#### Returns:

Error code: [NPP\\_NULL\\_POINTER\\_ERROR](#) if *hpQuantRawTable* is 0.

### 7.50.2.9 NppStatus nppiQuantInvTableInit\_JPEG\_8u16u (const Npp8u \* *hpQuantRawTable*, Npp16u \* *hpQuantFwdRawTable*)

Initializes a quantization table for [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R\(\)](#).

The [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) method uses a quantization table in a 16-bit format allowing for faster processing. In addition it converts the data order from zigzag layout to original row-order layout. Typically raw quantization tables are stored in zigzag format.

This method is a host method and consumes and produces host data. I.e. the pointers passed to this function must be host pointers. The resulting table needs to be transferred to device memory in order to be used with [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_C1R\(\)](#) function.

**Parameters:**

*hpQuantRawTable* Raw quantization table.

*hpQuantFwdRawTable* Inverse quantization table.

**Returns:**

[NPP\\_NULL\\_POINTER\\_ERROR](#) pQuantRawTable or pQuantFwdRawTable is 0.

## 7.51 Labeling and Segmentation

Pixel labeling and image segmentation operations.

### Modules

- [GraphCut](#)

### Typedefs

- typedef struct [NppiGraphcutState](#) [NppiGraphcutState](#)

#### 7.51.1 Detailed Description

Pixel labeling and image segmentation operations.

#### 7.51.2 Typedef Documentation

##### 7.51.2.1 typedef struct [NppiGraphcutState](#) [NppiGraphcutState](#)

## 7.52 GraphCut

### Graphcut

- [NppStatus nppiGraphcutGetSize](#) ([NppiSize](#) oSize, int \*pBufSize)  
*Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.*
- [NppStatus nppiGraphcut8GetSize](#) ([NppiSize](#) oSize, int \*pBufSize)  
*Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.*
- [NppStatus nppiGraphcutInitAlloc](#) ([NppiSize](#) oSize, [NppiGraphcutState](#) \*\*ppState, [Npp8u](#) \*pDeviceMem)  
*Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.*
- [NppStatus nppiGraphcut8InitAlloc](#) ([NppiSize](#) oSize, [NppiGraphcutState](#) \*\*ppState, [Npp8u](#) \*pDeviceMem)  
*Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.*
- [NppStatus nppiGraphcutFree](#) ([NppiGraphcutState](#) \*pState)  
*Frees the additional resources of the graph-cut state structure.*
- [NppStatus nppiGraphcut\\_32s8u](#) ([Npp32s](#) \*pTerminals, [Npp32s](#) \*pLeftTransposed, [Npp32s](#) \*pRightTransposed, [Npp32s](#) \*pTop, [Npp32s](#) \*pBottom, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit signed integer edge capacities).*
- [NppStatus nppiGraphcut8\\_32s8u](#) ([Npp32s](#) \*pTerminals, [Npp32s](#) \*pLeftTransposed, [Npp32s](#) \*pRightTransposed, [Npp32s](#) \*pTop, [Npp32s](#) \*pTopLeft, [Npp32s](#) \*pTopRight, [Npp32s](#) \*pBottom, [Npp32s](#) \*pBottomLeft, [Npp32s](#) \*pBottomRight, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit signed integer edge capacities).*
- [NppStatus nppiGraphcut\\_32f8u](#) ([Npp32f](#) \*pTerminals, [Npp32f](#) \*pLeftTransposed, [Npp32f](#) \*pRightTransposed, [Npp32f](#) \*pTop, [Npp32f](#) \*pBottom, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit float edge capacities).*
- [NppStatus nppiGraphcut8\\_32f8u](#) ([Npp32f](#) \*pTerminals, [Npp32f](#) \*pLeftTransposed, [Npp32f](#) \*pRightTransposed, [Npp32f](#) \*pTop, [Npp32f](#) \*pTopLeft, [Npp32f](#) \*pTopRight, [Npp32f](#) \*pBottom, [Npp32f](#) \*pBottomLeft, [Npp32f](#) \*pBottomRight, int nStep, int nTransposedStep, [NppiSize](#) size, [Npp8u](#) \*pLabel, int nLabelStep, [NppiGraphcutState](#) \*pState)  
*Graphcut of a flow network (32bit float edge capacities).*

## 7.52.1 Function Documentation

**7.52.1.1** `NppStatus nppiGraphcut8_32f8u (Npp32f * pTerminals, Npp32f * pLeftTransposed, Npp32f * pRightTransposed, Npp32f * pTop, Npp32f * pTopLeft, Npp32f * pTopRight, Npp32f * pBottom, Npp32f * pBottomLeft, Npp32f * pBottomRight, int nStep, int nTransposedStep, NppiSize size, Npp8u * pLabel, int nLabelStep, NppiGraphcutState * pState)`

Graphcut of a flow network (32bit float edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 8-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( `terminals(x) = source(x) - sink(x)` ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcut8GetSize\(\)](#).

Parameters:

***pTerminals*** Pointer to differences of terminal edge capacities (`terminal(x) = source(x) - sink(x)`)

***pLeftTransposed*** Pointer to transposed left edge capacities (`left(0,*)` must be 0)

***pRightTransposed*** Pointer to transposed right edge capacities (`right(width-1,*)` must be 0)

***pTop*** Pointer to top edge capacities (`top(*,0)` must be 0)

***pTopLeft*** Pointer to top left edge capacities (`topleft(*,0)` & `topleft(0,*)` must be 0)

***pTopRight*** Pointer to top right edge capacities (`topright(*,0)` & `topright(width-1,*)` must be 0)

***pBottom*** Pointer to bottom edge capacities (`bottom(*,height-1)` must be 0)

***pBottomLeft*** Pointer to bottom left edge capacities (`bottomleft(*,height-1)` && `bottomleft(0,*)` must be 0)

***pBottomRight*** Pointer to bottom right edge capacities (`bottomright(*,height-1)` && `bottomright(width-1,*)` must be 0)

***nStep*** Step in bytes between any pair of sequential rows of edge capacities

***nTransposedStep*** Step in bytes between any pair of sequential rows of tranposed edge capacities

***size*** Graph size

***pLabel*** Pointer to destination label image

***nLabelStep*** Step in bytes between any pair of sequential rows of label image

***pState*** Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcut8InitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.52.1.2 NppStatus nppiGraphcut8\_32s8u (Npp32s \* *pTerminals*, Npp32s \* *pLeftTransposed*, Npp32s \* *pRightTransposed*, Npp32s \* *pTop*, Npp32s \* *pTopLeft*, Npp32s \* *pTopRight*, Npp32s \* *pBottom*, Npp32s \* *pBottomLeft*, Npp32s \* *pBottomRight*, int *nStep*, int *nTransposedStep*, NppiSize *size*, Npp8u \* *pLabel*, int *nLabelStep*, NppiGraphcutState \* *pState*)**

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 8-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( *terminals*(*x*) = *source*(*x*) - *sink*(*x*) ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example *left*(0,\*) == 0). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcut8GetSize\(\)](#).

Parameters:

***pTerminals*** Pointer to differences of terminal edge capacities (*terminal*(*x*) = *source*(*x*) - *sink*(*x*))

***pLeftTransposed*** Pointer to transposed left edge capacities (*left*(0,\*) must be 0)

***pRightTransposed*** Pointer to transposed right edge capacities (*right*(width-1,\*) must be 0)

***pTop*** Pointer to top edge capacities (*top*(\*,0) must be 0)

***pTopLeft*** Pointer to top left edge capacities (*topleft*(\*,0) & *topleft*(0,\*) must be 0)

***pTopRight*** Pointer to top right edge capacities (*topright*(\*,0) & *topright*(width-1,\*) must be 0)

***pBottom*** Pointer to bottom edge capacities (*bottom*(\*,height-1) must be 0)

***pBottomLeft*** Pointer to bottom left edge capacities (*bottomleft*(\*,height-1) && *bottomleft*(0,\*) must be 0)

***pBottomRight*** Pointer to bottom right edge capacities (*bottomright*(\*,height-1) && *bottomright*(width-1,\*) must be 0)

***nStep*** Step in bytes between any pair of sequential rows of edge capacities

***nTransposedStep*** Step in bytes between any pair of sequential rows of transposed edge capacities

***size*** Graph size

***pLabel*** Pointer to destination label image

***nLabelStep*** Step in bytes between any pair of sequential rows of label image

***pState*** Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcut8InitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.52.1.3 NppStatus nppiGraphcut8GetSize (NppiSize *oSize*, int \* *pBufSize*)**

Calculates the size of the temporary buffer for graph-cut with 8 neighborhood labeling.

See also:

[nppiGraphcut8InitAlloc\(\)](#), [nppiGraphcut8\\_32s8u\(\)](#).

**Parameters:**

*oSize* Graph size.

*pBufSize* Pointer to variable that returns the size of the temporary buffer.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

#### 7.52.1.4 NppStatus nppiGraphcut8InitAlloc (NppiSize *oSize*, NppiGraphcutState \*\* *ppState*, Npp8u \* *pDeviceMem*)

Allocates and initializes the graph-cut state structure and additional resources for graph-cut with 8 neighborhood labeling.

See also:

[nppiGraphcut8\\_32s8u\(\)](#), [nppiGraphcut8GetSize\(\)](#).

**Parameters:**

*oSize* Graph size

*ppState* Pointer to pointer to graph-cut state structure.

*pDeviceMem* to the sufficient amount of device memory. The CUDA runtime or NPP memory allocators must be used to allocate this memory. The minimum amount of device memory required to run graph-cut on a for a specific image size is computed by [nppiGraphcut8GetSize\(\)](#).

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

#### 7.52.1.5 NppStatus nppiGraphcut\_32f8u (Npp32f \* *pTerminals*, Npp32f \* *pLeftTransposed*, Npp32f \* *pRightTransposed*, Npp32f \* *pTop*, Npp32f \* *pBottom*, int *nStep*, int *nTransposedStep*, NppiSize *size*, Npp8u \* *pLabel*, int *nLabelStep*, NppiGraphcutState \* *pState*)

Graphcut of a flow network (32bit float edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 4-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( `terminals(x) = source(x) - sink(x)` ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example `left(0,*) == 0`). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

See also:

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcutGetSize\(\)](#).

Parameters:

***pTerminals*** Pointer to differences of terminal edge capacities (terminal(x) = source(x) - sink(x))  
***pLeftTransposed*** Pointer to transposed left edge capacities (left(0,\*) must be 0)  
***pRightTransposed*** Pointer to transposed right edge capacities (right(width-1,\*) must be 0)  
***pTop*** Pointer to top edge capacities (top(\*,0) must be 0)  
***pBottom*** Pointer to bottom edge capacities (bottom(\*,height-1) must be 0)  
***nStep*** Step in bytes between any pair of sequential rows of edge capacities  
***nTransposedStep*** Step in bytes between any pair of sequential rows of transposed edge capacities  
***size*** Graph size  
***pLabel*** Pointer to destination label image  
***nLabelStep*** Step in bytes between any pair of sequential rows of label image  
***pState*** Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcutInitAlloc\(\)](#).

Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.52.1.6** `NppStatus nppiGraphcut_32s8u (Npp32s * pTerminals, Npp32s * pLeftTransposed, Npp32s * pRightTransposed, Npp32s * pTop, Npp32s * pBottom, int nStep, int nTransposedStep, NppiSize size, Npp8u * pLabel, int nLabelStep, NppiGraphcutState * pState)`

Graphcut of a flow network (32bit signed integer edge capacities).

The function computes the minimal cut (graphcut) of a 2D regular 4-connected graph. The inputs are the capacities of the horizontal (in transposed form), vertical and terminal (source and sink) edges. The capacities to source and sink are stored as capacity differences in the terminals array ( terminals(x) = source(x) - sink(x) ). The implementation assumes that the edge capacities for boundary edges that would connect to nodes outside the specified domain are set to 0 (for example left(0,\*) == 0). If this is not fulfilled the computed labeling may be wrong! The computed binary labeling is encoded as unsigned 8bit values (0 and >0).

See also:

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcutFree\(\)](#), [nppiGraphcutGetSize\(\)](#).

Parameters:

***pTerminals*** Pointer to differences of terminal edge capacities (terminal(x) = source(x) - sink(x))  
***pLeftTransposed*** Pointer to transposed left edge capacities (left(0,\*) must be 0)  
***pRightTransposed*** Pointer to transposed right edge capacities (right(width-1,\*) must be 0)  
***pTop*** Pointer to top edge capacities (top(\*,0) must be 0)  
***pBottom*** Pointer to bottom edge capacities (bottom(\*,height-1) must be 0)  
***nStep*** Step in bytes between any pair of sequential rows of edge capacities



*nTransposedStep* Step in bytes between any pair of sequential rows of tranposed edge capacities

*size* Graph size

*pLabel* Pointer to destination label image

*nLabelStep* Step in bytes between any pair of sequential rows of label image

*pState* Pointer to graph-cut state structure. This structure must be initialized allocated and initialized using [nppiGraphcutInitAlloc\(\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.52.1.7 NppStatus nppiGraphcutFree (NppiGraphcutState \* pState)

Frees the additional resources of the graph-cut state structure.

**See also:**

[nppiGraphcutInitAlloc](#)  
[nppiGraphcut8InitAlloc](#)

**Parameters:**

*pState* Pointer to graph-cut state structure.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning  
 NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value  
 NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pState pointer is NULL

### 7.52.1.8 NppStatus nppiGraphcutGetSize (NppiSize oSize, int \* pBufSize)

Calculates the size of the temporary buffer for graph-cut with 4 neighborhood labeling.

**See also:**

[nppiGraphcutInitAlloc\(\)](#), [nppiGraphcut\\_32s8u\(\)](#).

**Parameters:**

*oSize* Graph size.

*pBufSize* Pointer to variable that returns the size of the temporary buffer.

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning  
 NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value  
 NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

**7.52.1.9 NppStatus nppiGraphcutInitAlloc (NppiSize *oSize*, NppiGraphcutState \*\* *ppState*, Npp8u \* *pDeviceMem*)**

Initializes graph-cut state structure and allocates additional resources for graph-cut with 8 neighborhood labeling.

**See also:**

[nppiGraphcut\\_32s8u\(\)](#), [nppiGraphcutGetSize\(\)](#).

**Parameters:**

*oSize* Graph size

*ppState* Pointer to pointer to graph-cut state structure.

*pDeviceMem* pDeviceMem to the sufficient amount of device memory. The CUDA runtime or NPP memory allocators must be used to allocate this memory. The minimum amount of device memory required to run graph-cut on a for a specific image size is computed by [nppiGraphcutGetSize\(\)](#).

**Returns:**

NPP\_SUCCESS Indicates no error. Any other value indicates an error or a warning

NPP\_SIZE\_ERROR Indicates an error condition if any image dimension has zero or negative value

NPP\_NULL\_POINTER\_ERROR Indicates an error condition if pBufSize pointer is NULL

## 7.53 Data Exchange and Initialization

Primitives for initializing, copying and converting image data.

### Modules

- [Set](#)

*Primitives for setting pixels to a specific value.*

- [Copy](#)
- [Convert](#)
- [Scale](#)
- [Copy Constant Border](#)
- [Copy Replicate Border](#)
- [Copy Wrap Border](#)
- [Copy Sub-Pixel](#)
- [Duplicate Channel](#)
- [Transpose](#)
- [Swap Channels](#)

### 7.53.1 Detailed Description

Primitives for initializing, copying and converting image data.

## 7.54 Set

Primitives for setting pixels to a specific value.

### Set

Set all pixels within the ROI to a specific value.

- **NppStatus nppiSet\_8s\_C1R** (const **Npp8s** nValue, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit image set.*
- **NppStatus nppiSet\_8s\_C2R** (const **Npp8s** aValue[2], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit two-channel image set.*
- **NppStatus nppiSet\_8s\_C3R** (const **Npp8s** aValue[3], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit three-channel image set.*
- **NppStatus nppiSet\_8s\_C4R** (const **Npp8s** aValue[4], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit four-channel image set.*
- **NppStatus nppiSet\_8s\_AC4R** (const **Npp8s** aValue[3], **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit four-channel image set ignoring alpha channel.*
- **NppStatus nppiSet\_8u\_C1R** (const **Npp8u** nValue, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit unsigned image set.*
- **NppStatus nppiSet\_8u\_C2R** (const **Npp8u** aValue[2], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 8-bit unsigned image set.*
- **NppStatus nppiSet\_8u\_C3R** (const **Npp8u** aValue[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 8-bit unsigned image set.*
- **NppStatus nppiSet\_8u\_C4R** (const **Npp8u** aValue[4], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned image set.*
- **NppStatus nppiSet\_8u\_AC4R** (const **Npp8u** aValue[3], **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_16u\_C1R** (const **Npp16u** nValue, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit unsigned image set.*

- **NppStatus nppiSet\_16u\_C2R** (const **Npp16u** aValue[2], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_C3R** (const **Npp16u** aValue[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_C4R** (const **Npp16u** aValue[4], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image set.*
- **NppStatus nppiSet\_16u\_AC4R** (const **Npp16u** aValue[3], **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_16s\_C1R** (const **Npp16s** nValue, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit image set.*
- **NppStatus nppiSet\_16s\_C2R** (const **Npp16s** aValue[2], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 16-bit image set.*
- **NppStatus nppiSet\_16s\_C3R** (const **Npp16s** aValue[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 16-bit image set.*
- **NppStatus nppiSet\_16s\_C4R** (const **Npp16s** aValue[4], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit image set.*
- **NppStatus nppiSet\_16s\_AC4R** (const **Npp16s** aValue[3], **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_16sc\_C1R** (const **Npp16sc** oValue, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit complex integer image set.*
- **NppStatus nppiSet\_16sc\_C2R** (const **Npp16sc** aValue[2], **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit complex integer two-channel image set.*
- **NppStatus nppiSet\_16sc\_C3R** (const **Npp16sc** aValue[3], **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit complex integer three-channel image set.*
- **NppStatus nppiSet\_16sc\_C4R** (const **Npp16sc** aValue[4], **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*16-bit complex integer four-channel image set.*

- `NppStatus nppiSet_16sc_AC4R` (const `Npp16sc` aValue[3], `Npp16sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*16-bit complex integer four-channel image set ignoring alpha.*

- `NppStatus nppiSet_32s_C1R` (const `Npp32s` nValue, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*32-bit image set.*

- `NppStatus nppiSet_32s_C2R` (const `Npp32s` aValue[2], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*2 channel 32-bit image set.*

- `NppStatus nppiSet_32s_C3R` (const `Npp32s` aValue[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 32-bit image set.*

- `NppStatus nppiSet_32s_C4R` (const `Npp32s` aValue[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 32-bit image set.*

- `NppStatus nppiSet_32s_AC4R` (const `Npp32s` aValue[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 32-bit image set method, not affecting Alpha channel.*

- `NppStatus nppiSet_32u_C1R` (const `Npp32u` nValue, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*32-bit unsigned image set.*

- `NppStatus nppiSet_32u_C2R` (const `Npp32u` aValue[2], `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*2 channel 32-bit unsigned image set.*

- `NppStatus nppiSet_32u_C3R` (const `Npp32u` aValue[3], `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*3 channel 32-bit unsigned image set.*

- `NppStatus nppiSet_32u_C4R` (const `Npp32u` aValue[4], `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 32-bit unsigned image set.*

- `NppStatus nppiSet_32u_AC4R` (const `Npp32u` aValue[3], `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*4 channel 32-bit unsigned image set method, not affecting Alpha channel.*

- `NppStatus nppiSet_32sc_C1R` (const `Npp32sc` oValue, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 32-bit complex integer image set.*

- **NppStatus nppiSet\_32sc\_C2R** (const **Npp32sc** aValue[2], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Two channel 32-bit complex integer image set.*
- **NppStatus nppiSet\_32sc\_C3R** (const **Npp32sc** aValue[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit complex integer image set.*
- **NppStatus nppiSet\_32sc\_C4R** (const **Npp32sc** aValue[4], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit complex integer image set.*
- **NppStatus nppiSet\_32sc\_AC4R** (const **Npp32sc** aValue[3], **Npp32sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit complex integer four-channel image set ignoring alpha.*
- **NppStatus nppiSet\_32f\_C1R** (const **Npp32f** nValue, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_C2R** (const **Npp32f** aValue[2], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*2 channel 32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_C3R** (const **Npp32f** aValue[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_C4R** (const **Npp32f** aValue[4], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit floating point image set.*
- **NppStatus nppiSet\_32f\_AC4R** (const **Npp32f** aValue[3], **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit floating point image set method, not affecting Alpha channel.*
- **NppStatus nppiSet\_32fc\_C1R** (const **Npp32fc** oValue, **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 32-bit complex image set.*
- **NppStatus nppiSet\_32fc\_C2R** (const **Npp32fc** aValue[2], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Two channel 32-bit complex image set.*
- **NppStatus nppiSet\_32fc\_C3R** (const **Npp32fc** aValue[3], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit complex image set.*
- **NppStatus nppiSet\_32fc\_C4R** (const **Npp32fc** aValue[4], **Npp32fc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit complex image set.*

- `NppStatus nppiSet_32fc_AC4R` (const `Npp32fc` aValue[3], `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*32-bit complex four-channel image set ignoring alpha.*

## Masked Set

The masked set primitives have an additional "mask image" input.

The mask controls which pixels within the ROI are set. For details see [Masked Operation](#).

- `NppStatus nppiSet_8u_C1MR` (`Npp8u` nValue, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 8-bit unsigned image set.*

- `NppStatus nppiSet_8u_C3MR` (const `Npp8u` aValue[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 3 channel 8-bit unsigned image set.*

- `NppStatus nppiSet_8u_C4MR` (const `Npp8u` aValue[4], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 4 channel 8-bit unsigned image set.*

- `NppStatus nppiSet_8u_AC4MR` (const `Npp8u` aValue[3], `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.*

- `NppStatus nppiSet_16u_C1MR` (`Npp16u` nValue, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 16-bit unsigned image set.*

- `NppStatus nppiSet_16u_C3MR` (const `Npp16u` aValue[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 3 channel 16-bit unsigned image set.*

- `NppStatus nppiSet_16u_C4MR` (const `Npp16u` aValue[4], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 4 channel 16-bit unsigned image set.*

- `NppStatus nppiSet_16u_AC4MR` (const `Npp16u` aValue[3], `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.*

- `NppStatus nppiSet_16s_C1MR` (`Npp16s` nValue, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 16-bit image set.*

- `NppStatus nppiSet_16s_C3MR` (const `Npp16s` aValue[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked 3 channel 16-bit image set.*



- `NppStatus nppiSet_16s_C4MR` (const `Npp16s` aValue[4], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set.*
- `NppStatus nppiSet_16s_AC4MR` (const `Npp16s` aValue[3], `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_32s_C1MR` (`Npp32s` nValue, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 32-bit image set.*
- `NppStatus nppiSet_32s_C3MR` (const `Npp32s` aValue[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 3 channel 32-bit image set.*
- `NppStatus nppiSet_32s_C4MR` (const `Npp32s` aValue[4], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit image set.*
- `NppStatus nppiSet_32s_AC4MR` (const `Npp32s` aValue[3], `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 16-bit image set method, not affecting Alpha channel.*
- `NppStatus nppiSet_32f_C1MR` (`Npp32f` nValue, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 32-bit floating point image set.*
- `NppStatus nppiSet_32f_C3MR` (const `Npp32f` aValue[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 3 channel 32-bit floating point image set.*
- `NppStatus nppiSet_32f_C4MR` (const `Npp32f` aValue[4], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit floating point image set.*
- `NppStatus nppiSet_32f_AC4MR` (const `Npp32f` aValue[3], `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)  
*Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.*

## Channel Set

The select-channel set primitives set a single color channel in multi-channel images to a given value.

The channel is selected by adjusting the pDst pointer to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiSet_8u_C3CR` (`Npp8u` nValue, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*3 channel 8-bit unsigned image set affecting only single channel.*

- **NppStatus nppiSet\_8u\_C4CR** (**Npp8u** nValue, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_16u\_C3CR** (**Npp16u** nValue, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 16-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_16u\_C4CR** (**Npp16u** nValue, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_16s\_C3CR** (**Npp16s** nValue, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 16-bit signed image set affecting only single channel.*
- **NppStatus nppiSet\_16s\_C4CR** (**Npp16s** nValue, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit signed image set affecting only single channel.*
- **NppStatus nppiSet\_32s\_C3CR** (**Npp32s** nValue, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 32-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_32s\_C4CR** (**Npp32s** nValue, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit unsigned image set affecting only single channel.*
- **NppStatus nppiSet\_32f\_C3CR** (**Npp32f** nValue, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*3 channel 32-bit floating point image set affecting only single channel.*
- **NppStatus nppiSet\_32f\_C4CR** (**Npp32f** nValue, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 32-bit floating point image set affecting only single channel.*

### 7.54.1 Detailed Description

Primitives for setting pixels to a specific value.

### 7.54.2 Function Documentation

#### 7.54.2.1 NppStatus nppiSet\_16s\_AC4MR (const Npp16s aValue[3], Npp16s \*pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \*pMask, int nMaskStep)

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

#### Parameters:

**aValue** The pixel-value to be set.

**pDst** Destination-Image Pointer.

**nDstStep** Destination-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

**pMask** Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.2 NppStatus nppiSet\_16s\_AC4R (const Npp16s aValue[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.3 NppStatus nppiSet\_16s\_C1MR (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 16-bit image set.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.4 NppStatus nppiSet\_16s\_C1R (const Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit image set.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.5 NppStatus nppiSet\_16s\_C2R (const Npp16s aValue[2], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

2 channel 16-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.6 NppStatus nppiSet\_16s\_C3CR (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 16-bit signed image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.7 NppStatus nppiSet\_16s\_C3MR (const Npp16s aValue[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 3 channel 16-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.8 NppStatus nppiSet\_16s\_C3R (const Npp16s aValue[3], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

3 channel 16-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.9 NppStatus nppiSet\_16s\_C4CR (Npp16s nValue, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit signed image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.10 NppStatus nppiSet\_16s\_C4MR (const Npp16s aValue[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 4 channel 16-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.11 NppStatus nppiSet\_16s\_C4R (const Npp16s aValue[4], Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.12 NppStatus nppiSet\_16sc\_AC4R (const Npp16sc aValue[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer four-channel image set ignoring alpha.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.13 NppStatus nppiSet\_16sc\_C1R (const Npp16sc oValue, Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer image set.

**Parameters:**

*oValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.14 NppStatus nppiSet\_16sc\_C2R (const Npp16sc aValue[2], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer two-channel image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.15 NppStatus nppiSet\_16sc\_C3R (const Npp16sc aValue[3], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer three-channel image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.16 NppStatus nppiSet\_16sc\_C4R (const Npp16sc aValue[4], Npp16sc \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit complex integer four-channel image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.17** `NppStatus nppiSet_16u_AC4MR (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 16-bit unsigned image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.18** `NppStatus nppiSet_16u_AC4R (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.19** `NppStatus nppiSet_16u_C1MR (Npp16u nValue, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 16-bit unsigned image set.

**Parameters:**

*nValue* The pixel-value to be set.



*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.54.2.20 NppStatus nppiSet\_16u\_C1R (const Npp16u *nValue*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

16-bit unsigned image set.

**Parameters:**

*nValue* The pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.54.2.21 NppStatus nppiSet\_16u\_C2R (const Npp16u *aValue*[2], Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

2 channel 16-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.54.2.22 NppStatus nppiSet\_16u\_C3CR (Npp16u *nValue*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 16-bit unsigned image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.23** `NppStatus nppiSet_16u_C3MR (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 3 channel 16-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.24** `NppStatus nppiSet_16u_C3R (const Npp16u aValue[3], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 16-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.25** `NppStatus nppiSet_16u_C4CR (Npp16u nValue, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.26** `NppStatus nppiSet_16u_C4MR (const Npp16u aValue[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 16-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.27** `NppStatus nppiSet_16u_C4R (const Npp16u aValue[4], Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.28** `NppStatus nppiSet_32f_AC4MR (const Npp32f aValue[3], Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit floating point image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.29 NppStatus nppiSet\_32f\_AC4R (const Npp32f aValue[3], Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 32-bit floating point image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.30 NppStatus nppiSet\_32f\_C1MR (Npp32f nValue, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked 32-bit floating point image set.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.31 NppStatus nppiSet\_32f\_C1R (const Npp32f *nValue*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

32-bit floating point image set.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.32 NppStatus nppiSet\_32f\_C2R (const Npp32f *aValue*[2], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

2 channel 32-bit floating point image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.33 NppStatus nppiSet\_32f\_C3CR (Npp32f *nValue*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 32-bit floating point image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.54.2.34 **NppStatus nppiSet\_32f\_C3MR** (const Npp32f *aValue*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked 3 channel 32-bit floating point image set.

##### Parameters:

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.54.2.35 **NppStatus nppiSet\_32f\_C3R** (const Npp32f *aValue*[3], Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

3 channel 32-bit floating point image set.

##### Parameters:

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.54.2.36 **NppStatus nppiSet\_32f\_C4CR** (Npp32f *nValue*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit floating point image set affecting only single channel.

##### Parameters:

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.37** `NppStatus nppiSet_32f_C4MR (const Npp32f aValue[4], Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 32-bit floating point image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.38** `NppStatus nppiSet_32f_C4R (const Npp32f aValue[4], Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.39** `NppStatus nppiSet_32fc_AC4R (const Npp32fc aValue[3], Npp32fc * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit complex four-channel image set ignoring alpha.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.40 NppStatus nppiSet\_32fc\_C1R (const Npp32fc oValue, Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 32-bit complex image set.

**Parameters:**

*oValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.41 NppStatus nppiSet\_32fc\_C2R (const Npp32fc aValue[2], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Two channel 32-bit complex image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.42 NppStatus nppiSet\_32fc\_C3R (const Npp32fc aValue[3], Npp32fc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 32-bit complex image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



#### 7.54.2.43 **NppStatus nppiSet\_32fc\_C4R** (const Npp32fc *aValue*[4], Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit complex image set.

##### Parameters:

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.54.2.44 **NppStatus nppiSet\_32s\_AC4MR** (const Npp32s *aValue*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked 4 channel 16-bit image set method, not affecting Alpha channel.

##### Parameters:

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.54.2.45 **NppStatus nppiSet\_32s\_AC4R** (const Npp32s *aValue*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image set method, not affecting Alpha channel.

##### Parameters:

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.46** `NppStatus nppiSet_32s_C1MR (Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 32-bit image set.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.47** `NppStatus nppiSet_32s_C1R (const Npp32s nValue, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit image set.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.48** `NppStatus nppiSet_32s_C2R (const Npp32s aValue[2], Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

2 channel 32-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.49 NppStatus nppiSet\_32s\_C3CR (Npp32s *nValue*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 32-bit unsigned image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.  
*pDst* [Select-Channel Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.50 NppStatus nppiSet\_32s\_C3MR (const Npp32s *aValue*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked 3 channel 32-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.51 NppStatus nppiSet\_32s\_C3R (const Npp32s *aValue*[3], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 32-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.52 NppStatus nppiSet\_32s\_C4CR (Npp32s *nValue*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 32-bit unsigned image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.  
*pDst* [Select-Channel Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.53 NppStatus nppiSet\_32s\_C4MR (const Npp32s *aValue*[4], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked 4 channel 32-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.54 NppStatus nppiSet\_32s\_C4R (const Npp32s *aValue*[4], Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 32-bit image set.

**Parameters:**

*aValue* The pixel-value to be set.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.55 NppStatus nppiSet\_32sc\_AC4R (const Npp32sc aValue[3], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit complex integer four-channel image set ignoring alpha.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.56 NppStatus nppiSet\_32sc\_C1R (const Npp32sc oValue, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 32-bit complex integer image set.

**Parameters:**

*oValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.57 NppStatus nppiSet\_32sc\_C2R (const Npp32sc aValue[2], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Two channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.58 NppStatus nppiSet\_32sc\_C3R (const Npp32sc aValue[3], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.59 NppStatus nppiSet\_32sc\_C4R (const Npp32sc aValue[4], Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 32-bit complex integer image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.60 NppStatus nppiSet\_32u\_AC4R (const Npp32u aValue[3], Npp32u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 32-bit unsigned image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.61 NppStatus nppiSet\_32u\_C1R (const Npp32u *nValue*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

32-bit unsigned image set.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.62 NppStatus nppiSet\_32u\_C2R (const Npp32u *aValue*[2], Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

2 channel 32-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.63 NppStatus nppiSet\_32u\_C3R (const Npp32u *aValue*[3], Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 32-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.64 NppStatus nppiSet\_32u\_C4R (const Npp32u *aValue*[4], Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 32-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.65 NppStatus nppiSet\_8s\_AC4R (const Npp8s *aValue*[3], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit four-channel image set ignoring alpha channel.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.66 NppStatus nppiSet\_8s\_C1R (const Npp8s *nValue*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit image set.

**Parameters:**

*nValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.54.2.67 NppStatus nppiSet\_8s\_C2R (const Npp8s *aValue*[2], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit two-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.68 NppStatus nppiSet\_8s\_C3R (const Npp8s *aValue*[3], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit three-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.69 NppStatus nppiSet\_8s\_C4R (const Npp8s *aValue*[4], Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit four-channel image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.70** `NppStatus nppiSet_8u_AC4MR (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 8-bit unsigned image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.71** `NppStatus nppiSet_8u_AC4R (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set method, not affecting Alpha channel.

**Parameters:**

*aValue* The pixel-value to be set.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.72** `NppStatus nppiSet_8u_C1MR (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 8-bit unsigned image set.

**Parameters:**

*nValue* The pixel value to be set.  
*pDst* Pointer [Destination-Image Pointer](#).  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.73 NppStatus nppiSet\_8u\_C1R (const Npp8u *nValue*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit unsigned image set.

**Parameters:**

*nValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.74 NppStatus nppiSet\_8u\_C2R (const Npp8u *aValue*[2], Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

2 channel 8-bit unsigned image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.75 NppStatus nppiSet\_8u\_C3CR (Npp8u *nValue*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

3 channel 8-bit unsigned image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.76** `NppStatus nppiSet_8u_C3MR (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 3 channel 8-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.77** `NppStatus nppiSet_8u_C3R (const Npp8u aValue[3], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

3 channel 8-bit unsigned image set.

**Parameters:**

*aValue* The pixel value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.78** `NppStatus nppiSet_8u_C4CR (Npp8u nValue, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set affecting only single channel.

**Parameters:**

*nValue* The pixel-value to be set.

*pDst* Select-Channel Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.54.2.79** `NppStatus nppiSet_8u_C4MR (const Npp8u aValue[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked 4 channel 8-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.54.2.80** `NppStatus nppiSet_8u_C4R (const Npp8u aValue[4], Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 8-bit unsigned image set.

**Parameters:**

*aValue* The pixel-value to be set.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.55 Copy

### Copy

Copy pixels from one image to another.

- **NppStatus nppiCopy\_8s\_C1R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit image copy.*
- **NppStatus nppiCopy\_8s\_C2R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Two-channel 8-bit image copy.*
- **NppStatus nppiCopy\_8s\_C3R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 8-bit image copy.*
- **NppStatus nppiCopy\_8s\_C4R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit image copy.*
- **NppStatus nppiCopy\_8s\_AC4R** (const **Npp8s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit image copy, ignoring alpha channel.*
- **NppStatus nppiCopy\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*8-bit unsigned image copy.*
- **NppStatus nppiCopy\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 8-bit unsigned image copy.*
- **NppStatus nppiCopy\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned image copy.*
- **NppStatus nppiCopy\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 8-bit unsigned image copy, not affecting Alpha channel.*
- **NppStatus nppiCopy\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit unsigned image copy.*
- **NppStatus nppiCopy\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 16-bit unsigned image copy.*

- **NppStatus nppiCopy\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image copy.*
- **NppStatus nppiCopy\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit unsigned image copy, not affecting Alpha channel.*
- **NppStatus nppiCopy\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit image copy.*
- **NppStatus nppiCopy\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 16-bit image copy.*
- **NppStatus nppiCopy\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit image copy.*
- **NppStatus nppiCopy\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*4 channel 16-bit image copy, not affecting Alpha.*
- **NppStatus nppiCopy\_16sc\_C1R** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*16-bit complex image copy.*
- **NppStatus nppiCopy\_16sc\_C2R** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Two-channel 16-bit complex image copy.*
- **NppStatus nppiCopy\_16sc\_C3R** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit complex image copy.*
- **NppStatus nppiCopy\_16sc\_C4R** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit complex image copy.*
- **NppStatus nppiCopy\_16sc\_AC4R** (const **Npp16sc** \*pSrc, int nSrcStep, **Npp16sc** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit complex image copy, ignoring alpha.*
- **NppStatus nppiCopy\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit image copy.*
- **NppStatus nppiCopy\_32s\_C3R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit image copy.*

- `NppStatus nppiCopy_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit image copy.*
- `NppStatus nppiCopy_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit image copy, not affecting Alpha.*
- `NppStatus nppiCopy_32sc_C1R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_C2R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Two-channel 32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_C3R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three-channel 32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_C4R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit complex image copy.*
- `NppStatus nppiCopy_32sc_AC4R` (const `Npp32sc` \*pSrc, int nSrcStep, `Npp32sc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four-channel 32-bit complex image copy, ignoring alpha.*
- `NppStatus nppiCopy_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit floating point image copy.*
- `NppStatus nppiCopy_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit floating point image copy.*
- `NppStatus nppiCopy_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image copy.*
- `NppStatus nppiCopy_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*4 channel 32-bit floating point image copy, not affecting Alpha.*
- `NppStatus nppiCopy_32fc_C1R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*32-bit floating-point complex image copy.*
- `NppStatus nppiCopy_32fc_C2R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)



*Two-channel 32-bit floating-point complex image copy.*

- `NppStatus nppiCopy_32fc_C3R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit floating-point complex image copy.*

- `NppStatus nppiCopy_32fc_C4R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit floating-point complex image copy.*

- `NppStatus nppiCopy_32fc_AC4R` (const `Npp32fc` \*pSrc, int nSrcStep, `Npp32fc` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit floating-point complex image copy, ignoring alpha.*

## Masked Copy

The masked copy primitives have an additional "mask image" input.

The mask controls which pixels within the ROI are copied. For details see [Masked Operation](#).

- `NppStatus nppiCopy_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_C3MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation three channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_C4MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation four channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_AC4MR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation four channel 8-bit unsigned image copy, ignoring alpha.*

- `NppStatus nppiCopy_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_C3MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation three channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_C4MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation four channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_AC4MR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 16-bit unsigned image copy, ignoring alpha.

- `NppStatus nppiCopy_16s_C1MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* 16-bit signed image copy.

- `NppStatus nppiCopy_16s_C3MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* three channel 16-bit signed image copy.

- `NppStatus nppiCopy_16s_C4MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 16-bit signed image copy.

- `NppStatus nppiCopy_16s_AC4MR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 16-bit signed image copy, ignoring alpha.

- `NppStatus nppiCopy_32s_C1MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* 32-bit signed image copy.

- `NppStatus nppiCopy_32s_C3MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* three channel 32-bit signed image copy.

- `NppStatus nppiCopy_32s_C4MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 32-bit signed image copy.

- `NppStatus nppiCopy_32s_AC4MR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 32-bit signed image copy, ignoring alpha.

- `NppStatus nppiCopy_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* 32-bit float image copy.

- `NppStatus nppiCopy_32f_C3MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* three channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_C4MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 32-bit float image copy.

- `NppStatus nppiCopy_32f_AC4MR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` \*pMask, int nMaskStep)

*Masked Operation* four channel 32-bit float image copy, ignoring alpha.

## Channel Copy

The channel copy primitives copy a single color channel from a multi-channel source image to any other color channel in a multi-channel destination image.

The channel is selected by adjusting the respective image pointers to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiCopy_8u_C3CR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 8-bit unsigned image copy for three-channel images.*

- `NppStatus nppiCopy_8u_C4CR` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 8-bit unsigned image copy for four-channel images.*

- `NppStatus nppiCopy_16s_C3CR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 16-bit signed image copy for three-channel images.*

- `NppStatus nppiCopy_16s_C4CR` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 16-bit signed image copy for four-channel images.*

- `NppStatus nppiCopy_16u_C3CR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 16-bit unsigned image copy for three-channel images.*

- `NppStatus nppiCopy_16u_C4CR` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 16-bit unsigned image copy for four-channel images.*

- `NppStatus nppiCopy_32s_C3CR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 32-bit signed image copy for three-channel images.*

- `NppStatus nppiCopy_32s_C4CR` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 32-bit signed image copy for four-channel images.*

- `NppStatus nppiCopy_32f_C3CR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 32-bit float image copy for three-channel images.*

- `NppStatus nppiCopy_32f_C4CR` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Select-channel 32-bit float image copy for four-channel images.*

## Extract Channel Copy

The channel extract primitives copy a single color channel from a multi-channel source image to single-channel destination image.

The channel is selected by adjusting the source image pointer to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiCopy_8u_C3C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel to single-channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_C4C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel to single-channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_16s_C3C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel to single-channel 16-bit signed image copy.*

- `NppStatus nppiCopy_16s_C4C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel to single-channel 16-bit signed image copy.*

- `NppStatus nppiCopy_16u_C3C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel to single-channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_C4C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel to single-channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_32s_C3C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel to single-channel 32-bit signed image copy.*

- `NppStatus nppiCopy_32s_C4C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel to single-channel 32-bit signed image copy.*

- `NppStatus nppiCopy_32f_C3C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three-channel to single-channel 32-bit float image copy.*

- `NppStatus nppiCopy_32f_C4C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four-channel to single-channel 32-bit float image copy.*

## Insert Channel Copy

The channel insert primitives copy a single-channel source image into one of the color channels in a multi-channel destination image.

The channel is selected by adjusting the destination image pointer to point to the desired color channel (see [Channel-of-Interest API](#)).

- `NppStatus nppiCopy_8u_C1C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_8u_C1C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 8-bit unsigned image copy.*

- `NppStatus nppiCopy_16s_C1C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 16-bit signed image copy.*

- `NppStatus nppiCopy_16s_C1C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 16-bit signed image copy.*

- `NppStatus nppiCopy_16u_C1C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_16u_C1C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 16-bit unsigned image copy.*

- `NppStatus nppiCopy_32s_C1C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 32-bit signed image copy.*

- `NppStatus nppiCopy_32s_C1C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 32-bit signed image copy.*

- `NppStatus nppiCopy_32f_C1C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to three-channel 32-bit float image copy.*

- `NppStatus nppiCopy_32f_C1C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single-channel to four-channel 32-bit float image copy.*

## Packed-to-Planar Copy

Split a packed multi-channel image into a planar image.

E.g. copy the three channels of an RGB image into three separate single-channel images.

- `NppStatus nppiCopy_8u_C3P3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*const aDst[3], int nDstStep, `NppiSize` oSizeROI)

*Three-channel 8-bit unsigned packed to planar image copy.*

- `NppStatus nppiCopy_8u_C4P4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 8-bit unsigned packed to planar image copy.*

- `NppStatus nppiCopy_16s_C3P3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*const aDst[3], int nDstStep, `NppiSize` oSizeROI)

*Three-channel 16-bit signed packed to planar image copy.*

- `NppStatus nppiCopy_16s_C4P4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 16-bit signed packed to planar image copy.*

- `NppStatus nppiCopy_16u_C3P3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*const aDst[3], int nDstStep, `NppiSize` oSizeROI)

*Three-channel 16-bit unsigned packed to planar image copy.*

- `NppStatus nppiCopy_16u_C4P4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 16-bit unsigned packed to planar image copy.*

- `NppStatus nppiCopy_32s_C3P3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*const aDst[3], int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit signed packed to planar image copy.*

- `NppStatus nppiCopy_32s_C4P4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit signed packed to planar image copy.*

- `NppStatus nppiCopy_32f_C3P3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*const aDst[3], int nDstStep, `NppiSize` oSizeROI)

*Three-channel 32-bit float packed to planar image copy.*

- `NppStatus nppiCopy_32f_C4P4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*const aDst[4], int nDstStep, `NppiSize` oSizeROI)

*Four-channel 32-bit float packed to planar image copy.*

## Planar-to-Packed Copy

Combine multiple image planes into a packed multi-channel image.

E.g. copy three single-channel images into a single 3-channel image.

- **NppStatus nppiCopy\_8u\_P3C3R** (const **Npp8u** \*const aSrc[3], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 8-bit unsigned planar to packed image copy.*
- **NppStatus nppiCopy\_8u\_P4C4R** (const **Npp8u** \*const aSrc[4], int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit unsigned planar to packed image copy.*
- **NppStatus nppiCopy\_16u\_P3C3R** (const **Npp16u** \*const aSrc[3], int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit unsigned planar to packed image copy.*
- **NppStatus nppiCopy\_16u\_P4C4R** (const **Npp16u** \*const aSrc[4], int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit unsigned planar to packed image copy.*
- **NppStatus nppiCopy\_16s\_P3C3R** (const **Npp16s** \*const aSrc[3], int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit signed planar to packed image copy.*
- **NppStatus nppiCopy\_16s\_P4C4R** (const **Npp16s** \*const aSrc[4], int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit signed planar to packed image copy.*
- **NppStatus nppiCopy\_32s\_P3C3R** (const **Npp32s** \*const aSrc[3], int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 32-bit signed planar to packed image copy.*
- **NppStatus nppiCopy\_32s\_P4C4R** (const **Npp32s** \*const aSrc[4], int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit signed planar to packed image copy.*
- **NppStatus nppiCopy\_32f\_P3C3R** (const **Npp32f** \*const aSrc[3], int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three-channel 32-bit float planar to packed image copy.*
- **NppStatus nppiCopy\_32f\_P4C4R** (const **Npp32f** \*const aSrc[4], int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit float planar to packed image copy.*

## 7.55.1 Function Documentation

### 7.55.1.1 **NppStatus nppiCopy\_16s\_AC4MR** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, int nMaskStep)

**Masked Operation** four channel 16-bit signed image copy, ignoring alpha.

#### Parameters:

**pSrc** Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.2 NppStatus nppiCopy\_16s\_AC4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 16-bit image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.3 NppStatus nppiCopy\_16s\_C1C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to three-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.55.1.4 NppStatus nppiCopy\_16s\_C1C4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to four-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.5 NppStatus nppiCopy\_16s\_C1MR (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \* pMask, int nMaskStep)**

Masked Operation 16-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.6 NppStatus nppiCopy\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI)**

16-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.7 **NppStatus nppiCopy\_16s\_C3C1R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel to single-channel 16-bit signed image copy.

#### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.8 **NppStatus nppiCopy\_16s\_C3CR** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 16-bit signed image copy for three-channel images.

#### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.9 **NppStatus nppiCopy\_16s\_C3MR** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation three channel 16-bit signed image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.10 NppStatus nppiCopy\_16s\_C3P3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \*const *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 16-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.11 NppStatus nppiCopy\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 16-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.12 NppStatus nppiCopy\_16s\_C4C1R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel to single-channel 16-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.13 `NppStatus nppiCopy_16s_C4CR (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 16-bit signed image copy for four-channel images.

#### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.14 `NppStatus nppiCopy_16s_C4MR (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation four channel 16-bit signed image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.15 `NppStatus nppiCopy_16s_C4P4R (const Npp16s * pSrc, int nSrcStep, Npp16s * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit signed packed to planar image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.16 NppStatus nppiCopy\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 16-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.17 NppStatus nppiCopy\_16s\_P3C3R (const Npp16s \*const *aSrc*[3], int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 16-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.18 NppStatus nppiCopy\_16s\_P4C4R (const Npp16s \*const *aSrc*[4], int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 16-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.19 NppStatus nppiCopy\_16sc\_AC4R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 16-bit complex image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.20 NppStatus nppiCopy\_16sc\_C1R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.21 NppStatus nppiCopy\_16sc\_C2R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 16-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.22 NppStatus nppiCopy\_16sc\_C3R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 16-bit complex image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.23 NppStatus nppiCopy\_16sc\_C4R (const Npp16sc \* *pSrc*, int *nSrcStep*, Npp16sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 16-bit complex image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.24 NppStatus nppiCopy\_16u\_AC4MR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

[Masked Operation](#) four channel 16-bit unsigned image copy, ignoring alpha.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.25 NppStatus nppiCopy\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

4 channel 16-bit unsigned image copy, not affecting Alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.26 NppStatus nppiCopy\_16u\_C1C3R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to three-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.27 NppStatus nppiCopy\_16u\_C1C4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Single-channel to four-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.55.1.28** `NppStatus nppiCopy_16u_C1MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) 16-bit unsigned image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.29** `NppStatus nppiCopy_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

16-bit unsigned image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.30** `NppStatus nppiCopy_16u_C3C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* [Select-Channel Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.55.1.31 `NppStatus nppiCopy_16u_C3CR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Select-channel 16-bit unsigned image copy for three-channel images.

#### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.32 `NppStatus nppiCopy_16u_C3MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

Masked Operation three channel 16-bit unsigned image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.33 `NppStatus nppiCopy_16u_C3P3R (const Npp16u * pSrc, int nSrcStep, Npp16u * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 16-bit unsigned packed to planar image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.34 NppStatus nppiCopy\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.35 NppStatus nppiCopy\_16u\_C4C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel to single-channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.36 NppStatus nppiCopy\_16u\_C4CR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI)**

Select-channel 16-bit unsigned image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.37** `NppStatus nppiCopy_16u_C4MR (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.38** `NppStatus nppiCopy_16u_C4P4R (const Npp16u * pSrc, int nSrcStep, Npp16u * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 16-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.39** `NppStatus nppiCopy_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 16-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.55.1.40 NppStatus nppiCopy\_16u\_P3C3R (const Npp16u \*const aSrc[3], int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)

Three-channel 16-bit unsigned planar to packed image copy.

#### Parameters:

*aSrc* Planar [Source-Planar-Image Pointer Array](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.55.1.41 NppStatus nppiCopy\_16u\_P4C4R (const Npp16u \*const aSrc[4], int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)

Four-channel 16-bit unsigned planar to packed image copy.

#### Parameters:

*aSrc* Planar [Source-Planar-Image Pointer Array](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.55.1.42 NppStatus nppiCopy\_32f\_AC4MR (const Npp32f \*pSrc, int nSrcStep, Npp32f \*pDst, int nDstStep, NppiSize oSizeROI, const Npp8u \*pMask, int nMaskStep)

[Masked Operation](#) four channel 32-bit float image copy, ignoring alpha.

#### Parameters:

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.43 NppStatus nppiCopy\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 32-bit floating point image copy, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.44 NppStatus nppiCopy\_32f\_C1C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to three-channel 32-bit float image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.45 NppStatus nppiCopy\_32f\_C1C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to four-channel 32-bit float image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.46** `NppStatus nppiCopy_32f_C1MR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) 32-bit float image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.47** `NppStatus nppiCopy_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

32-bit floating point image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.48** `NppStatus nppiCopy_32f_C3C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 32-bit float image copy.

**Parameters:**

*pSrc* [Select-Channel Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.55.1.49 **NppStatus nppiCopy\_32f\_C3CR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 32-bit float image copy for three-channel images.

##### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.55.1.50 **NppStatus nppiCopy\_32f\_C3MR** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation three channel 32-bit float image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.55.1.51 **NppStatus nppiCopy\_32f\_C3P3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit float packed to planar image copy.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes



**7.55.1.52 NppStatus nppiCopy\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 32-bit floating point image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.53 NppStatus nppiCopy\_32f\_C4C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel to single-channel 32-bit float image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.54 NppStatus nppiCopy\_32f\_C4CR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Select-channel 32-bit float image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.55** `NppStatus nppiCopy_32f_C4MR (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 32-bit float image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.56** `NppStatus nppiCopy_32f_C4P4R (const Npp32f * pSrc, int nSrcStep, Npp32f * const aDst[4], int nDstStep, NppiSize oSizeROI)`

Four-channel 32-bit float packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.57** `NppStatus nppiCopy_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

4 channel 32-bit floating point image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.58 NppStatus nppiCopy\_32f\_P3C3R (const Npp32f \*const *aSrc*[3], int *nSrcStep*, Npp32f \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit float planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Planar-Image Pointer Array](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.59 NppStatus nppiCopy\_32f\_P4C4R (const Npp32f \*const *aSrc*[4], int *nSrcStep*, Npp32f \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit float planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Planar-Image Pointer Array](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.60 NppStatus nppiCopy\_32fc\_AC4R (const Npp32fc \**pSrc*, int *nSrcStep*, Npp32fc \**pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit floating-point complex image copy, ignoring alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.61 NppStatus nppiCopy\_32fc\_C1R (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

32-bit floating-point complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.62 NppStatus nppiCopy\_32fc\_C2R (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 32-bit floating-point complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.63 NppStatus nppiCopy\_32fc\_C3R (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit floating-point complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.64 NppStatus nppiCopy\_32fc\_C4R (const Npp32fc \* *pSrc*, int *nSrcStep*, Npp32fc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point complex image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.65 NppStatus nppiCopy\_32s\_AC4MR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation four channel 32-bit signed image copy, ignoring alpha.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.66 NppStatus nppiCopy\_32s\_AC4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 32-bit image copy, not affecting Alpha.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.67 **NppStatus nppiCopy\_32s\_C1C3R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to three-channel 32-bit signed image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.68 **NppStatus nppiCopy\_32s\_C1C4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single-channel to four-channel 32-bit signed image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.69 **NppStatus nppiCopy\_32s\_C1MR** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation 32-bit signed image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.70 NppStatus nppiCopy\_32s\_C1R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

32-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.71 NppStatus nppiCopy\_32s\_C3C1R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel to single-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.72 NppStatus nppiCopy\_32s\_C3CR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Select-channel 32-bit signed image copy for three-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.73** `NppStatus nppiCopy_32s_C3MR (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) three channel 32-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.74** `NppStatus nppiCopy_32s_C3P3R (const Npp32s * pSrc, int nSrcStep, Npp32s * const aDst[3], int nDstStep, NppiSize oSizeROI)`

Three-channel 32-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.75** `NppStatus nppiCopy_32s_C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 32-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.55.1.76 NppStatus nppiCopy\_32s\_C4C1R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel to single-channel 32-bit signed image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.77 NppStatus nppiCopy\_32s\_C4CR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Select-channel 32-bit signed image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.78 NppStatus nppiCopy\_32s\_C4MR (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)**

Masked Operation four channel 32-bit signed image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.79 NppStatus nppiCopy\_32s\_C4P4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \*const *aDst*[4], int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit signed packed to planar image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.80 NppStatus nppiCopy\_32s\_C4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 32-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.81 NppStatus nppiCopy\_32s\_P3C3R (const Npp32s \*const *aSrc*[3], int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.82 NppStatus nppiCopy\_32s\_P4C4R (const Npp32s \*const aSrc[4], int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 32-bit signed planar to packed image copy.

**Parameters:**

*aSrc* Planar Source-Planar-Image Pointer Array.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.83 NppStatus nppiCopy\_32sc\_AC4R (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 32-bit complex image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.84 NppStatus nppiCopy\_32sc\_C1R (const Npp32sc \* pSrc, int nSrcStep, Npp32sc \* pDst, int nDstStep, NppiSize oSizeROI)**

32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.85 NppStatus nppiCopy\_32sc\_C2R (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.86 NppStatus nppiCopy\_32sc\_C3R (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.87 NppStatus nppiCopy\_32sc\_C4R (const Npp32sc \* *pSrc*, int *nSrcStep*, Npp32sc \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit complex image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.88 NppStatus nppiCopy\_8s\_AC4R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit image copy, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.89 NppStatus nppiCopy\_8s\_C1R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.90 NppStatus nppiCopy\_8s\_C2R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Two-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.91** `NppStatus nppiCopy_8s_C3R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.92** `NppStatus nppiCopy_8s_C4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Four-channel 8-bit image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.93** `NppStatus nppiCopy_8u_AC4MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) four channel 8-bit unsigned image copy, ignoring alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.94 NppStatus nppiCopy\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

4 channel 8-bit unsigned image copy, not affecting Alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.95 NppStatus nppiCopy\_8u\_C1C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to three-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.96 NppStatus nppiCopy\_8u\_C1C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single-channel to four-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.97** `NppStatus nppiCopy_8u_C1MR (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, int nMaskStep)`

[Masked Operation](#) 8-bit unsigned image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.98** `NppStatus nppiCopy_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

8-bit unsigned image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.99** `NppStatus nppiCopy_8u_C3C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three-channel to single-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* [Select-Channel Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.55.1.100 **NppStatus nppiCopy\_8u\_C3CR** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Select-channel 8-bit unsigned image copy for three-channel images.

#### Parameters:

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.101 **NppStatus nppiCopy\_8u\_C3MR** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

Masked Operation three channel 8-bit unsigned image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.55.1.102 **NppStatus nppiCopy\_8u\_C3P3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* const *aDst*[3], int *nDstStep*, NppiSize *oSizeROI*)

Three-channel 8-bit unsigned packed to planar image copy.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*aDst* Destination-Planar-Image Pointer Array.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.103 NppStatus nppiCopy\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.104 NppStatus nppiCopy\_8u\_C4C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel to single-channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.105 NppStatus nppiCopy\_8u\_C4CR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Select-channel 8-bit unsigned image copy for four-channel images.

**Parameters:**

*pSrc* Select-Channel Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Select-Channel Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.55.1.106** **NppStatus nppiCopy\_8u\_C4MR** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u \* *pMask*, int *nMaskStep*)

[Masked Operation](#) four channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.107** **NppStatus nppiCopy\_8u\_C4P4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \*const *aDst*[4], int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned packed to planar image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*aDst* [Destination-Planar-Image Pointer Array](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.108** **NppStatus nppiCopy\_8u\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

4 channel 8-bit unsigned image copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.109 NppStatus nppiCopy\_8u\_P3C3R (const Npp8u \*const aSrc[3], int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three-channel 8-bit unsigned planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Image Pointer](#).  
*nSrcStep* [Source-Planar-Image Pointer Array](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.55.1.110 NppStatus nppiCopy\_8u\_P4C4R (const Npp8u \*const aSrc[4], int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four-channel 8-bit unsigned planar to packed image copy.

**Parameters:**

*aSrc* Planar [Source-Planar-Image Pointer Array](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.56 Convert

### Convert to Increase Bit-Depth

The integer conversion methods do not involve any scaling.

Also, even when increasing the bit-depth loss of information may occur:

- When converting integers (e.g. Npp32u) to float (e.g. Npp32f) integervalue not accurately representable by the float are rounded to the closest floating-point value.
- When converting signed integers to unsigned integers all negative values are lost (saturated to 0).
- **NppStatus nppiConvert\_8u16u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit unsigned to 16-bit unsigned conversion.*
- **NppStatus nppiConvert\_8u16u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 8-bit unsigned to 16-bit unsigned conversion.*
- **NppStatus nppiConvert\_8u16u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 16-bit unsigned conversion.*
- **NppStatus nppiConvert\_8u16u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_8u16s\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit unsigned to 16-bit signed conversion.*
- **NppStatus nppiConvert\_8u16s\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 8-bit unsigned to 16-bit signed conversion.*
- **NppStatus nppiConvert\_8u16s\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 16-bit signed conversion.*
- **NppStatus nppiConvert\_8u16s\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_8u32s\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit unsigned to 32-bit signed conversion.*
- **NppStatus nppiConvert\_8u32s\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI)

*Three channel 8-bit unsigned to 32-bit signed conversion.*

- `NppStatus nppiConvert_8u32s_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned to 32-bit signed conversion.*

- `NppStatus nppiConvert_8u32s_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.*

- `NppStatus nppiConvert_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit unsigned to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_8u32f_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit unsigned to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.*

- `NppStatus nppiConvert_8s32s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit signed to 32-bit signed conversion.*

- `NppStatus nppiConvert_8s32s_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit signed to 32-bit signed conversion.*

- `NppStatus nppiConvert_8s32s_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit signed to 32-bit signed conversion.*

- `NppStatus nppiConvert_8s32s_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit signed to 32-bit signed conversion, not affecting Alpha.*

- `NppStatus nppiConvert_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit signed to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit signed to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit signed to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit signed to 32-bit floating-point conversion, not affecting Alpha.*
- `NppStatus nppiConvert_16u32s_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit unsigned to 32-bit signed conversion.*
- `NppStatus nppiConvert_16u32s_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit unsigned to 32-bit signed conversion.*
- `NppStatus nppiConvert_16u32s_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit unsigned to 32-bit signed conversion.*
- `NppStatus nppiConvert_16u32s_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit unsigned to 32-bit signed conversion, not affecting Alpha.*
- `NppStatus nppiConvert_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit unsigned to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit unsigned to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_16u32f_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit unsigned to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_16u32f_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.*
- `NppStatus nppiConvert_16s32s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit signed to 32-bit signed conversion.*
- `NppStatus nppiConvert_16s32s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit signed to 32-bit signed conversion.*
- `NppStatus nppiConvert_16s32s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed to 32-bit signed conversion.*

- `NppStatus nppiConvert_16s32s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed to 32-bit signed conversion, not affecting Alpha.*
- `NppStatus nppiConvert_16s32f_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit signed to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_16s32f_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit signed to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_16s32f_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed to 32-bit floating-point conversion.*
- `NppStatus nppiConvert_16s32f_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed to 32-bit floating-point conversion, not affecting Alpha.*
- `NppStatus nppiConvert_8s8u_C1Rs` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit signed to 8-bit unsigned conversion with saturation.*
- `NppStatus nppiConvert_8s16u_C1Rs` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit signed to 16-bit unsigned conversion with saturation.*
- `NppStatus nppiConvert_8s16s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit signed to 16-bit signed conversion.*
- `NppStatus nppiConvert_8s32u_C1Rs` (const `Npp8s` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit signed to 32-bit unsigned conversion with saturation.*
- `NppStatus nppiConvert_16s16u_C1Rs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit signed to 16-bit unsigned conversion with saturation.*
- `NppStatus nppiConvert_16s32u_C1Rs` (const `Npp16s` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit signed to 32-bit unsigned conversion with saturation.*
- `NppStatus nppiConvert_16u32u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit unsigned to 32-bit unsigned conversion.*
- `NppStatus nppiConvert_32s32u_C1Rs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32u` \*pDst, int nDstStep, `NppiSize` oSizeROI)



*Single channel 32-bit signed to 32-bit unsigned conversion with saturation.*

- `NppStatus nppiConvert_32s32f_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 32-bit signed to 32-bit floating-point conversion.*

- `NppStatus nppiConvert_32u32f_C1R` (const `Npp32u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 32-bit unsigned to 32-bit floating-point conversion.*

## Convert to Decrease Bit-Depth

The integer conversion methods do not involve any scaling.

When converting floating-point values to integers the user may choose the most appropriate rounding-mode. Typically information is lost when converting to lower bit depth:

- All converted values are saturated to the destination type's range. E.g. any values larger than the largest value of the destination type are clamped to the destination's maximum.
- Converting floating-point values to integer also involves rounding, effectively losing all fractional value information in the process.

- `NppStatus nppiConvert_16u8u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit unsigned to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_16u8u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit unsigned to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_16u8u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit unsigned to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_16u8u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.*

- `NppStatus nppiConvert_16s8u_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_16s8u_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_16s8u_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed to 8-bit unsigned conversion.*

- **NppStatus nppiConvert\_16s8u\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_32s8u\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 32-bit signed to 8-bit unsigned conversion.*
- **NppStatus nppiConvert\_32s8u\_C3R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit signed to 8-bit unsigned conversion.*
- **NppStatus nppiConvert\_32s8u\_C4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit signed to 8-bit unsigned conversion.*
- **NppStatus nppiConvert\_32s8u\_AC4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_32s8s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Single channel 32-bit signed to 8-bit signed conversion.*
- **NppStatus nppiConvert\_32s8s\_C3R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit signed to 8-bit signed conversion.*
- **NppStatus nppiConvert\_32s8s\_C4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit signed to 8-bit signed conversion.*
- **NppStatus nppiConvert\_32s8s\_AC4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit signed to 8-bit signed conversion, not affecting Alpha.*
- **NppStatus nppiConvert\_8u8s\_C1RSfs** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 8-bit unsigned to 8-bit signed conversion.*
- **NppStatus nppiConvert\_16u8s\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 16-bit unsigned to 8-bit signed conversion.*
- **NppStatus nppiConvert\_16s8s\_C1RSfs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 16-bit signed to 8-bit signed conversion.*
- **NppStatus nppiConvert\_16u16s\_C1RSfs** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)

*Single channel 16-bit unsigned to 16-bit signed conversion.*

- `NppStatus nppiConvert_32u8u_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

*Single channel 32-bit unsigned to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_32u8s_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

*Single channel 32-bit unsigned to 8-bit signed conversion.*

- `NppStatus nppiConvert_32u16u_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

*Single channel 32-bit unsigned to 16-bit unsigned conversion.*

- `NppStatus nppiConvert_32u16s_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

*Single channel 32-bit unsigned to 16-bit signed conversion.*

- `NppStatus nppiConvert_32u32s_C1RSfs` (const `Npp32u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

*Single channel 32-bit unsigned to 32-bit signed conversion.*

- `NppStatus nppiConvert_32s16u_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

*Single channel 32-bit signed to 16-bit unsigned conversion.*

- `NppStatus nppiConvert_32s16s_C1RSfs` (const `Npp32s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode, int nScaleFactor)

*Single channel 32-bit signed to 16-bit signed conversion.*

- `NppStatus nppiConvert_32f8u_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

*Single channel 32-bit floating point to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_32f8u_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

*Three channel 32-bit floating point to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_32f8u_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

*Four channel 32-bit floating point to 8-bit unsigned conversion.*

- `NppStatus nppiConvert_32f8u_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

*Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.*

- `NppStatus nppiConvert_32f8s_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppRoundMode` eRoundMode)

*Single channel 32-bit floating point to 8-bit signed conversion.*

- **NppStatus** **nppiConvert\_32f8s\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Three channel 32-bit floating point to 8-bit signed conversion.*
- **NppStatus** **nppiConvert\_32f8s\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Four channel 32-bit floating point to 8-bit signed conversion.*
- **NppStatus** **nppiConvert\_32f8s\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Four channel 32-bit floating point to 8-bit signed conversion, not affecting Alpha.*
- **NppStatus** **nppiConvert\_32f16u\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Single channel 32-bit floating point to 16-bit unsigned conversion.*
- **NppStatus** **nppiConvert\_32f16u\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Three channel 32-bit floating point to 16-bit unsigned conversion.*
- **NppStatus** **nppiConvert\_32f16u\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Four channel 32-bit floating point to 16-bit unsigned conversion.*
- **NppStatus** **nppiConvert\_32f16u\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Four channel 32-bit floating point to 16-bit unsigned conversion, not affecting Alpha.*
- **NppStatus** **nppiConvert\_32f16s\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Single channel 32-bit floating point to 16-bit signed conversion.*
- **NppStatus** **nppiConvert\_32f16s\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Three channel 32-bit floating point to 16-bit signed conversion.*
- **NppStatus** **nppiConvert\_32f16s\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Four channel 32-bit floating point to 16-bit signed conversion.*
- **NppStatus** **nppiConvert\_32f16s\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode)  
*Four channel 32-bit floating point to 16-bit signed conversion.*
- **NppStatus** **nppiConvert\_32f8u\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 8-bit unsigned conversion.*
- **NppStatus** **nppiConvert\_32f8s\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp8s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 8-bit signed conversion.*

- **NppStatus nppiConvert\_32f16u\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 16-bit unsigned conversion.*
- **NppStatus nppiConvert\_32f16s\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 16-bit signed conversion.*
- **NppStatus nppiConvert\_32f32u\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 32-bit unsigned conversion.*
- **NppStatus nppiConvert\_32f32s\_C1RSfs** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppRoundMode** eRoundMode, int nScaleFactor)  
*Single channel 32-bit floating point to 32-bit signed conversion.*

### 7.56.1 Function Documentation

#### 7.56.1.1 **NppStatus nppiConvert\_16s16u\_C1Rs** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)

Single channel 16-bit signed to 16-bit unsigned conversion with saturation.

##### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.2 **NppStatus nppiConvert\_16s32f\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

Four channel 16-bit signed to 32-bit floating-point conversion, not affecting Alpha.

##### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.3 NppStatus nppiConvert\_16s32f\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Single channel 16-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.4 NppStatus nppiConvert\_16s32f\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 16-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.5 NppStatus nppiConvert\_16s32f\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 16-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.6 NppStatus nppiConvert\_16s32s\_AC4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.7 NppStatus nppiConvert\_16s32s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 16-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.8 NppStatus nppiConvert\_16s32s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 16-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.9 **NppStatus nppiConvert\_16s32s\_C4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.10 **NppStatus nppiConvert\_16s32u\_C1Rs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit signed to 32-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.11 **NppStatus nppiConvert\_16s8s\_C1RSfs** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)

Single channel 16-bit signed to 8-bit signed conversion.



**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Rounding Mode Parameter.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.12 NppStatus nppiConvert\_16s8u\_AC4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.13 NppStatus nppiConvert\_16s8u\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.14 **NppStatus nppiConvert\_16s8u\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit signed to 8-bit unsigned conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.15 **NppStatus nppiConvert\_16s8u\_C4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit signed to 8-bit unsigned conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.16 **NppStatus nppiConvert\_16u16s\_C1RSfs** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)

Single channel 16-bit unsigned to 16-bit signed conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Rounding Mode Parameter.  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.17** `NppStatus nppiConvert_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.18** `NppStatus nppiConvert_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.19** `NppStatus nppiConvert_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 16-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.20 NppStatus nppiConvert\_16u32f\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.21 NppStatus nppiConvert\_16u32s\_AC4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit unsigned to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.22 NppStatus nppiConvert\_16u32s\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 16-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.23 NppStatus nppiConvert\_16u32s\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 16-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.24 NppStatus nppiConvert\_16u32s\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.25 NppStatus nppiConvert\_16u32u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp32u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 16-bit unsigned to 32-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.26** `NppStatus nppiConvert_16u8s_C1RSfs (const Npp16u * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 16-bit unsigned to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Rounding Mode Parameter.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.27** `NppStatus nppiConvert_16u8u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.28** `NppStatus nppiConvert_16u8u_C1R (const Npp16u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.29 NppStatus nppiConvert\_16u8u\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Three channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.30 NppStatus nppiConvert\_16u8u\_C4R (const Npp16u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI)**

Four channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.31 NppStatus nppiConvert\_32f16s\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

Four channel 32-bit floating point to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.56.1.32 `NppStatus nppiConvert_32f16s_C1R (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 16-bit signed conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.56.1.33 `NppStatus nppiConvert_32f16s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 16-bit signed conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.56.1.34 `NppStatus nppiConvert_32f16s_C3R (const Npp32f * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 16-bit signed conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.35 NppStatus nppiConvert\_32f16s\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

Four channel 32-bit floating point to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.36 NppStatus nppiConvert\_32f16u\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)**

Four channel 32-bit floating point to 16-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.37** `NppStatus nppiConvert_32f16u_C1R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.38** `NppStatus nppiConvert_32f16u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.39** `NppStatus nppiConvert_32f16u_C3R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.40** `NppStatus nppiConvert_32f16u_C4R (const Npp32f * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.41** `NppStatus nppiConvert_32f32s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.42** `NppStatus nppiConvert_32f32u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp32u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 32-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.43** `NppStatus nppiConvert_32f8s_AC4R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 8-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.44** `NppStatus nppiConvert_32f8s_C1R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Single channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.45** `NppStatus nppiConvert_32f8s_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.46** `NppStatus nppiConvert_32f8s_C3R (const Npp32f * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.47 NppStatus nppiConvert\_32f8s\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*)**

Four channel 32-bit floating point to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.48 NppStatus nppiConvert\_32f8u\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*)**

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.49 NppStatus nppiConvert\_32f8u\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*)**

Single channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.50** `NppStatus nppiConvert_32f8u_C1RSfs (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.51** `NppStatus nppiConvert_32f8u_C3R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Three channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.52** `NppStatus nppiConvert_32f8u_C4R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode)`

Four channel 32-bit floating point to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Flag specifying how fractional float values are rounded to integer values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.53** `NppStatus nppiConvert_32s16s_C1RSfs (const Npp32s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Rounding Mode Parameter.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.54** `NppStatus nppiConvert_32s16u_C1RSfs (const Npp32s * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Rounding Mode Parameter.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.56.1.55** `NppStatus nppiConvert_32s32f_C1R (const Npp32s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 32-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.56** `NppStatus nppiConvert_32s32u_C1Rs (const Npp32s * pSrc, int nSrcStep, Npp32u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 32-bit signed to 32-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.57** `NppStatus nppiConvert_32s8s_AC4R (const Npp32s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 32-bit signed to 8-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.58 NppStatus nppiConvert\_32s8s\_C1R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit signed to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.59 NppStatus nppiConvert\_32s8s\_C3R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit signed to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.60 NppStatus nppiConvert\_32s8s\_C4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit signed to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.61** `NppStatus nppiConvert_32s8u_AC4R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.62** `NppStatus nppiConvert_32s8u_C1R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.63** `NppStatus nppiConvert_32s8u_C3R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.64 **NppStatus nppiConvert\_32s8u\_C4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit signed to 8-bit unsigned conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.56.1.65 **NppStatus nppiConvert\_32u16s\_C1RSfs** (const Npp32u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)

Single channel 32-bit unsigned to 16-bit signed conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Rounding Mode Parameter.  
*nScaleFactor* Integer Result Scaling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.56.1.66 **NppStatus nppiConvert\_32u16u\_C1RSfs** (const Npp32u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)

Single channel 32-bit unsigned to 16-bit unsigned conversion.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Rounding Mode Parameter.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.67 NppStatus nppiConvert\_32u32f\_C1R (const Npp32u \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 32-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.68 NppStatus nppiConvert\_32u32s\_C1RSfs (const Npp32u \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)**

Single channel 32-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eRoundMode* Rounding Mode Parameter.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.69** `NppStatus nppiConvert_32u8s_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Rounding Mode Parameter.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.70** `NppStatus nppiConvert_32u8u_C1RSfs (const Npp32u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppRoundMode eRoundMode, int nScaleFactor)`

Single channel 32-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Rounding Mode Parameter.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.71** `NppStatus nppiConvert_8s16s_C1R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit signed to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.72 NppStatus nppiConvert\_8s16u\_C1Rs (const Npp8s \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 16-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.73 NppStatus nppiConvert\_8s32f\_AC4R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit signed to 32-bit floating-point conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.56.1.74 NppStatus nppiConvert\_8s32f\_C1R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Single channel 8-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.75** `NppStatus nppiConvert_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.76** `NppStatus nppiConvert_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit signed to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.56.1.77 NppStatus nppiConvert\_8s32s\_AC4R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit signed to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.78 NppStatus nppiConvert\_8s32s\_C1R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.79 NppStatus nppiConvert\_8s32s\_C3R (const Npp8s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.80** `NppStatus nppiConvert_8s32s_C4R (const Npp8s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit signed to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.81** `NppStatus nppiConvert_8s32u_C1Rs (const Npp8s * pSrc, int nSrcStep, Npp32u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit signed to 32-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.82** `NppStatus nppiConvert_8s8u_C1Rs (const Npp8s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit signed to 8-bit unsigned conversion with saturation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.83** `NppStatus nppiConvert_8u16s_AC4R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.84** `NppStatus nppiConvert_8u16s_C1R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.85** `NppStatus nppiConvert_8u16s_C3R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.86 NppStatus nppiConvert\_8u16s\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.87 NppStatus nppiConvert\_8u16u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.88 NppStatus nppiConvert\_8u16u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.89 NppStatus nppiConvert\_8u16u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.90 NppStatus nppiConvert\_8u16u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.91 NppStatus nppiConvert\_8u32f\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.92 NppStatus nppiConvert\_8u32f\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.93 NppStatus nppiConvert\_8u32f\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.94 NppStatus nppiConvert\_8u32f\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 32-bit floating-point conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.56.1.95** `NppStatus nppiConvert_8u32s_AC4R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.96** `NppStatus nppiConvert_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.97** `NppStatus nppiConvert_8u32s_C3R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.98 NppStatus nppiConvert\_8u32s\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.56.1.99 NppStatus nppiConvert\_8u8s\_C1RSfs (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppRoundMode *eRoundMode*, int *nScaleFactor*)**

Single channel 8-bit unsigned to 8-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eRoundMode* Rounding Mode Parameter.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



## 7.57 Scale

### Scaled Bit-Depth Conversion

Scale bit-depth up and down.

To map source pixel `srcPixelValue` to destination pixel `dstPixelValue` the following equation is used:

$$\text{dstPixelValue} = \text{dstMinRangeValue} + \text{scaleFactor} * (\text{srcPixelValue} - \text{srcMinRangeValue})$$

where  $\text{scaleFactor} = (\text{dstMaxRangeValue} - \text{dstMinRangeValue}) / (\text{srcMaxRangeValue} - \text{srcMinRangeValue})$ .

For conversions between integer data types, the entire integer numeric range of the input data type is mapped onto the entire integer numeric range of the output data type.

For conversions to floating point data types the floating point data range is defined by the user supplied floating point values of `nMax` and `nMin` which are used as the `dstMaxRangeValue` and `dstMinRangeValue` respectively in the `scaleFactor` and `dstPixelValue` calculations and also as the saturation values to which output data is clamped.

When converting from floating-point values to integer values, `nMax` and `nMin` are used as the `srcMaxRangeValue` and `srcMinRangeValue` respectively in the `scaleFactor` and `dstPixelValue` calculations. Output values are saturated and clamped to the full output integer pixel value range.

- `NppStatus nppiScale_8u16u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiScale_8u16u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiScale_8u16u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned to 16-bit unsigned conversion.*
- `NppStatus nppiScale_8u16u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.*
- `NppStatus nppiScale_8u16s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit unsigned to 16-bit signed conversion.*
- `NppStatus nppiScale_8u16s_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Three channel 8-bit unsigned to 16-bit signed conversion.*
- `NppStatus nppiScale_8u16s_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned to 16-bit signed conversion.*
- `NppStatus nppiScale_8u16s_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.*

- `NppStatus nppiScale_8u32s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit unsigned to 32-bit signed conversion.*

- `NppStatus nppiScale_8u32s_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit unsigned to 32-bit signed conversion.*

- `NppStatus nppiScale_8u32s_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned to 32-bit signed conversion.*

- `NppStatus nppiScale_8u32s_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.*

- `NppStatus nppiScale_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Single channel 8-bit unsigned to 32-bit floating-point conversion.*

- `NppStatus nppiScale_8u32f_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Three channel 8-bit unsigned to 32-bit floating-point conversion.*

- `NppStatus nppiScale_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Four channel 8-bit unsigned to 32-bit floating-point conversion.*

- `NppStatus nppiScale_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.*

- `NppStatus nppiScale_16u8u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Single channel 16-bit unsigned to 8-bit unsigned conversion.*

- `NppStatus nppiScale_16u8u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Three channel 16-bit unsigned to 8-bit unsigned conversion.*

- `NppStatus nppiScale_16u8u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Four channel 16-bit unsigned to 8-bit unsigned conversion.*

- `NppStatus nppiScale_16u8u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.*

- `NppStatus nppiScale_16s8u_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Single channel 16-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiScale_16s8u_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Three channel 16-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiScale_16s8u_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Four channel 16-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiScale_16s8u_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.*

- `NppStatus nppiScale_32s8u_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Single channel 32-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiScale_32s8u_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Three channel 32-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiScale_32s8u_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Four channel 32-bit signed to 8-bit unsigned conversion.*

- `NppStatus nppiScale_32s8u_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `NppHintAlgorithm` hint)

*Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.*

- `NppStatus nppiScale_32f8u_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Single channel 32-bit floating point to 8-bit unsigned conversion.*

- `NppStatus nppiScale_32f8u_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Three channel 32-bit floating point to 8-bit unsigned conversion.*

- `NppStatus nppiScale_32f8u_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Four channel 32-bit floating point to 8-bit unsigned conversion.*

- `NppStatus nppiScale_32f8u_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, `Npp32f` nMin, `Npp32f` nMax)

*Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.*

## 7.57.1 Function Documentation

### 7.57.1.1 `NppStatus nppiScale_16s8u_AC4R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)`

Four channel 16-bit signed to 8-bit unsigned conversion, not affecting Alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.57.1.2 `NppStatus nppiScale_16s8u_C1R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)`

Single channel 16-bit signed to 8-bit unsigned conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.57.1.3 `NppStatus nppiScale_16s8u_C3R (const Npp16s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)`

Three channel 16-bit signed to 8-bit unsigned conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.4 NppStatus nppiScale\_16s8u\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)**

Four channel 16-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.5 NppStatus nppiScale\_16u8u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)**

Four channel 16-bit unsigned to 8-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.6 NppStatus nppiScale\_16u8u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)**

Single channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.7 NppStatus nppiScale\_16u8u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)**

Three channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.8 NppStatus nppiScale\_16u8u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)**

Four channel 16-bit unsigned to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.57.1.9 NppStatus nppiScale\_32f8u\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nMin*, Npp32f *nMax*)

Four channel 32-bit floating point to 8-bit unsigned conversion, not affecting Alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nMin* specifies the minimum saturation value to which every output value will be clamped.

*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, NPP\_SCALE\_RANGE\_ERROR indicates an error condition if  $nMax \leq nMin$ .

### 7.57.1.10 NppStatus nppiScale\_32f8u\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nMin*, Npp32f *nMax*)

Single channel 32-bit floating point to 8-bit unsigned conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nMin* specifies the minimum saturation value to which every output value will be clamped.

*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, NPP\_SCALE\_RANGE\_ERROR indicates an error condition if  $nMax \leq nMin$ .

### 7.57.1.11 NppStatus nppiScale\_32f8u\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, Npp32f *nMin*, Npp32f *nMax*)

Three channel 32-bit floating point to 8-bit unsigned conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nMin* specifies the minimum saturation value to which every output value will be clamped.

*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if  $nMax \leq nMin$ .

**7.57.1.12** `NppStatus nppiScale_32f8u_C4R (const Npp32f * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Four channel 32-bit floating point to 8-bit unsigned conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nMin* specifies the minimum saturation value to which every output value will be clamped.

*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if  $nMax \leq nMin$ .

**7.57.1.13** `NppStatus nppiScale_32s8u_AC4R (const Npp32s * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppHintAlgorithm hint)`

Four channel 32-bit signed to 8-bit unsigned conversion, not affecting Alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes



**7.57.1.14 NppStatus nppiScale\_32s8u\_C1R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)**

Single channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.15 NppStatus nppiScale\_32s8u\_C3R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)**

Three channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.16 NppStatus nppiScale\_32s8u\_C4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppHintAlgorithm *hint*)**

Four channel 32-bit signed to 8-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*hint* algorithm performance or accuracy selector, currently ignored

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.57.1.17 NppStatus nppiScale\_8u16s\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 16-bit signed conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.57.1.18 NppStatus nppiScale\_8u16s\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.57.1.19 NppStatus nppiScale\_8u16s\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.57.1.20 NppStatus nppiScale\_8u16s\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 16-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.57.1.21 NppStatus nppiScale\_8u16u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned to 16-bit unsigned conversion, not affecting Alpha.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.57.1.22 NppStatus nppiScale\_8u16u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned to 16-bit unsigned conversion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.57.1.23 `NppStatus nppiScale_8u16u_C3R (const Npp8u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 16-bit unsigned conversion.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.57.1.24 `NppStatus nppiScale_8u16u_C4R (const Npp8u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 16-bit unsigned conversion.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.57.1.25 `NppStatus nppiScale_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Four channel 8-bit unsigned to 32-bit floating-point conversion, not affecting Alpha.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nMin* specifies the minimum saturation value to which every output value will be clamped.  
*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if `nMax <= nMin`.

### 7.57.1.26 `NppStatus nppiScale_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Single channel 8-bit unsigned to 32-bit floating-point conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nMin* specifies the minimum saturation value to which every output value will be clamped.

*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if  $nMax \leq nMin$ .

### 7.57.1.27 `NppStatus nppiScale_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Three channel 8-bit unsigned to 32-bit floating-point conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nMin* specifies the minimum saturation value to which every output value will be clamped.

*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if  $nMax \leq nMin$ .

### 7.57.1.28 `NppStatus nppiScale_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nMin, Npp32f nMax)`

Four channel 8-bit unsigned to 32-bit floating-point conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nMin* specifies the minimum saturation value to which every output value will be clamped.

*nMax* specifies the maximum saturation value to which every output value will be clamped.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, `NPP_SCALE_RANGE_ERROR` indicates an error condition if  $nMax \leq nMin$ .

#### 7.57.1.29 `NppStatus nppiScale_8u32s_AC4R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 32-bit signed conversion, not affecting Alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.57.1.30 `NppStatus nppiScale_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Single channel 8-bit unsigned to 32-bit signed conversion.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.57.1.31** `NppStatus nppiScale_8u32s_C3R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Three channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.57.1.32** `NppStatus nppiScale_8u32s_C4R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned to 32-bit signed conversion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.58 Copy Constant Border

### CopyConstBorder

Methods for copying images and padding borders with a constant, user-specifiable color.

- `NppStatus nppiCopyConstBorder_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, `Npp8u` nValue)

*1 channel 8-bit unsigned integer image copy with constant border color.*

- `NppStatus nppiCopyConstBorder_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const `Npp8u` aValue[3])

*3 channel 8-bit unsigned integer image copy with constant border color.*

- `NppStatus nppiCopyConstBorder_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const `Npp8u` aValue[4])

*4 channel 8-bit unsigned integer image copy with constant border color.*

- `NppStatus nppiCopyConstBorder_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const `Npp8u` aValue[3])

*4 channel 8-bit unsigned integer image copy with constant border color with alpha channel unaffected.*

- `NppStatus nppiCopyConstBorder_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, `Npp16u` nValue)

*1 channel 16-bit unsigned integer image copy with constant border color.*

- `NppStatus nppiCopyConstBorder_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const `Npp16u` aValue[3])

*3 channel 16-bit unsigned integer image copy with constant border color.*

- `NppStatus nppiCopyConstBorder_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const `Npp16u` aValue[4])

*4 channel 16-bit unsigned integer image copy with constant border color.*

- `NppStatus nppiCopyConstBorder_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const `Npp16u` aValue[3])

*4 channel 16-bit unsigned integer image copy with constant border color with alpha channel unaffected.*

- `NppStatus nppiCopyConstBorder_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSrcSizeROI, `Npp16s` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, `Npp16s` nValue)

*1 channel 16-bit signed integer image copy with constant border color.*



- **NppStatus nppiCopyConstBorder\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16s** aValue[3])  
*3 channel 16-bit signed integer image copy with constant border color.*
- **NppStatus nppiCopyConstBorder\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16s** aValue[4])  
*4 channel 16-bit signed integer image copy with constant border color.*
- **NppStatus nppiCopyConstBorder\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp16s** aValue[3])  
*4 channel 16-bit signed integer image copy with constant border color with alpha channel unaffected.*
- **NppStatus nppiCopyConstBorder\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp32s** nValue)  
*1 channel 32-bit signed integer image copy with constant border color.*
- **NppStatus nppiCopyConstBorder\_32s\_C3R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32s** aValue[3])  
*3 channel 32-bit signed integer image copy with constant border color.*
- **NppStatus nppiCopyConstBorder\_32s\_C4R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32s** aValue[4])  
*4 channel 32-bit signed integer image copy with constant border color.*
- **NppStatus nppiCopyConstBorder\_32s\_AC4R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32s** aValue[3])  
*4 channel 32-bit signed integer image copy with constant border color with alpha channel unaffected.*
- **NppStatus nppiCopyConstBorder\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, **Npp32f** nValue)  
*1 channel 32-bit floating point image copy with constant border color.*
- **NppStatus nppiCopyConstBorder\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32f** aValue[3])  
*3 channel 32-bit floating point image copy with constant border color.*
- **NppStatus nppiCopyConstBorder\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const **Npp32f** aValue[4])  
*4 channel 32-bit floating point image copy with constant border color.*

- **NppStatus nppiCopyConstBorder\_32f\_AC4R** (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcSizeROI, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const [Npp32f](#) aValue[3])

*4 channel 32-bit floating point image copy with constant border color with alpha channel unaffected.*

## 7.58.1 Function Documentation

- 7.58.1.1 NppStatus nppiCopyConstBorder\_16s\_AC4R** (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcSizeROI, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const [Npp16s](#) aValue[3])

4 channel 16-bit signed integer image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSrcSizeROI](#) Size of the source region-of-interest.

[pDst](#) [Destination-Image Pointer](#).

[nDstStep](#) [Destination-Image Line Step](#).

[oDstSizeROI](#) Size of the destination region-of-interest.

[nTopBorderHeight](#) Height of top border.

[nLeftBorderWidth](#) Width of left border.

[aValue](#) Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.58.1.2 NppStatus nppiCopyConstBorder\_16s\_C1R** (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcSizeROI, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, [Npp16s](#) nValue)

1 channel 16-bit signed integer image copy with constant border color.

### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSrcSizeROI](#) Size of the source region of pixels.

[pDst](#) [Destination-Image Pointer](#).

[nDstStep](#) [Destination-Image Line Step](#).

[oDstSizeROI](#) Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

***nTopBorderHeight*** Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

***nLeftBorderWidth*** Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

***nValue*** The pixel value to be set for border pixels.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.3** `NppStatus nppiCopyConstBorder_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16s aValue[3])`

3 channel 16-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSizeROI*** Size of the source region-of-interest.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Size of the destination region-of-interest.

***nTopBorderHeight*** Height of top border.

***nLeftBorderWidth*** Width of left border.

***aValue*** Vector of the RGBA values of the border pixels to be set.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.4** `NppStatus nppiCopyConstBorder_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16s aValue[4])`

4 channel 16-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSizeROI*** Size of the source region-of-interest.

***pDst*** [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGBA values of the border pixels to be set.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.5** `NppStatus nppiCopyConstBorder_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp16u aValue[3])`

4 channel 16-bit unsigned integer image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.6** `NppStatus nppiCopyConstBorder_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp16u nValue)`

1 channel 16-bit unsigned integer image copy with constant border color.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region of pixels.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

***oDstSizeROI*** Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

***nTopBorderHeight*** Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

***nLeftBorderWidth*** Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

***nValue*** The pixel value to be set for border pixels.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.58.1.7 **NppStatus nppiCopyConstBorder\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*, const Npp16u *aValue*[3])

3 channel 16-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSizeROI*** Size of the source region-of-interest.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Size of the destination region-of-interest.

***nTopBorderHeight*** Height of top border.

***nLeftBorderWidth*** Width of left border.

***aValue*** Vector of the RGBA values of the border pixels to be set.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.58.1.8 **NppStatus nppiCopyConstBorder\_16u\_C4R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*, const Npp16u *aValue*[4])

4 channel 16-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.  
*aValue* Vector of the RGBA values of the border pixels to be set.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.58.1.9 **NppStatus nppiCopyConstBorder\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*, const Npp32f *aValue*[3])

4 channel 32-bit floating point image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.  
*aValue* Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.58.1.10 **NppStatus nppiCopyConstBorder\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*, Npp32f *nValue*)

1 channel 32-bit floating point image copy with constant border color.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region of pixels.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

*nValue* The pixel value to be set for border pixels.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.11** `NppStatus nppiCopyConstBorder_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32f aValue[3])`

3 channel 32-bit floating point image copy with constant border color.

See [nppiCopyConstBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGBA values of the border pixels to be set.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.12** `NppStatus nppiCopyConstBorder_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32f aValue[4])`

4 channel 32-bit floating point image copy with constant border color.

See [nppiCopyConstBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.  
*aValue* Vector of the RGBA values of the border pixels to be set.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.13** `NppStatus nppiCopyConstBorder_32s_AC4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32s aValue[3])`

4 channel 32-bit signed integer image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.  
*aValue* Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.14** `NppStatus nppiCopyConstBorder_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp32s nValue)`

1 channel 32-bit signed integer image copy with constant border color.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.



*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

*nValue* The pixel value to be set for border pixels.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.15** `NppStatus nppiCopyConstBorder_32s_C3R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32s aValue[3])`

3 channel 32-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGBA values of the border pixels to be set.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.16** `NppStatus nppiCopyConstBorder_32s_C4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp32s aValue[4])`

4 channel 32-bit signed integer image copy with constant border color.

See [nppiCopyConstBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.  
*aValue* Vector of the RGBA values of the border pixels to be set.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.17** `NppStatus nppiCopyConstBorder_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[3])`

4 channel 8-bit unsigned integer image copy with constant border color with alpha channel unaffected.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.  
*aValue* Vector of the RGB values of the border pixels. Because this method does not affect the destination image's alpha channel, only three components of the border color are needed.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.18** `NppStatus nppiCopyConstBorder_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, Npp8u nValue)`

1 channel 8-bit unsigned integer image copy with constant border color.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and constant border color (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the constant border color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

*nValue* The pixel value to be set for border pixels.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.19** `NppStatus nppiCopyConstBorder_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[3])`

3 channel 8-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGBA values of the border pixels to be set.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.58.1.20** `NppStatus nppiCopyConstBorder_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth, const Npp8u aValue[4])`

4 channel 8-bit unsigned integer image copy with constant border color.

See [nppiCopyConstBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

*aValue* Vector of the RGBA values of the border pixels to be set.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.59 Copy Replicate Border

### CopyReplicateBorder

Methods for copying images and padding borders with a replicates of the nearest source image pixel color.

- **NppStatus nppiCopyReplicateBorder\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*1 channel 8-bit unsigned integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*3 channel 8-bit unsigned integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 8-bit unsigned integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 8-bit unsigned integer image copy with nearest source image pixel color with alpha channel unaffected.*
- **NppStatus nppiCopyReplicateBorder\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*1 channel 16-bit unsigned integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*3 channel 16-bit unsigned integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 16-bit unsigned integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 16-bit unsigned image copy with nearest source image pixel color with alpha channel unaffected.*
- **NppStatus nppiCopyReplicateBorder\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*1 channel 16-bit signed integer image copy with nearest source image pixel color.*

- **NppStatus nppiCopyReplicateBorder\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*3 channel 16-bit signed integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*4 channel 16-bit signed integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*4 channel 16-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.*
- **NppStatus nppiCopyReplicateBorder\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*1 channel 32-bit signed integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_32s\_C3R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*3 channel 32-bit signed image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_32s\_C4R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*4 channel 32-bit signed integer image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_32s\_AC4R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*4 channel 32-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.*
- **NppStatus nppiCopyReplicateBorder\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*1 channel 32-bit floating point image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*3 channel 32-bit floating point image copy with nearest source image pixel color.*
- **NppStatus nppiCopyReplicateBorder\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)
 

*4 channel 32-bit floating point image copy with nearest source image pixel color.*

- [NppStatus nppiCopyReplicateBorder\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcSizeROI, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*4 channel 32-bit floating point image copy with nearest source image pixel color with alpha channel unaffected.*

## 7.59.1 Function Documentation

### 7.59.1.1 NppStatus nppiCopyReplicateBorder\_16s\_AC4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

[pSrc](#) [Source-Image Pointer](#).  
[nSrcStep](#) [Source-Image Line Step](#).  
[oSrcSizeROI](#) Size of the source region-of-interest.  
[pDst](#) [Destination-Image Pointer](#).  
[nDstStep](#) [Destination-Image Line Step](#).  
[oDstSizeROI](#) Size of the destination region-of-interest.  
[nTopBorderHeight](#) Height of top border.  
[nLeftBorderWidth](#) Width of left border.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.59.1.2 NppStatus nppiCopyReplicateBorder\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

1 channel 16-bit signed integer image copy with nearest source image pixel color.

#### Parameters:

[pSrc](#) [Source-Image Pointer](#).  
[nSrcStep](#) [Source-Image Line Step](#).  
[oSrcSizeROI](#) Size of the source region of pixels.  
[pDst](#) [Destination-Image Pointer](#).  
[nDstStep](#) [Destination-Image Line Step](#).  
[oDstSizeROI](#) Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).

***nTopBorderHeight*** Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

***nLeftBorderWidth*** Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.3 NppStatus nppiCopyReplicateBorder\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)**

3 channel 16-bit signed integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSizeROI*** Size of the source region-of-interest.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Size of the destination region-of-interest.

***nTopBorderHeight*** Height of top border.

***nLeftBorderWidth*** Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.4 NppStatus nppiCopyReplicateBorder\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)**

4 channel 16-bit signed integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSizeROI*** Size of the source region-of-interest.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Size of the destination region-of-interest.



*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.5 NppStatus nppiCopyReplicateBorder\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)**

4 channel 16-bit unsigned image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.6 NppStatus nppiCopyReplicateBorder\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u \* pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)**

1 channel 16-bit unsigned integer image copy with nearest source image pixel color.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

***nLeftBorderWidth*** Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.7** `NppStatus nppiCopyReplicateBorder_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

3 channel 16-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).  
***nSrcStep*** [Source-Image Line Step](#).  
***oSrcSizeROI*** Size of the source region-of-interest.  
***pDst*** [Destination-Image Pointer](#).  
***nDstStep*** [Destination-Image Line Step](#).  
***oDstSizeROI*** Size of the destination region-of-interest.  
***nTopBorderHeight*** Height of top border.  
***nLeftBorderWidth*** Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.8** `NppStatus nppiCopyReplicateBorder_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 16-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).  
***nSrcStep*** [Source-Image Line Step](#).  
***oSrcSizeROI*** Size of the source region-of-interest.  
***pDst*** [Destination-Image Pointer](#).  
***nDstStep*** [Destination-Image Line Step](#).  
***oDstSizeROI*** Size of the destination region-of-interest.  
***nTopBorderHeight*** Height of top border.  
***nLeftBorderWidth*** Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.59.1.9 NppStatus nppiCopyReplicateBorder\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 32-bit floating point image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.59.1.10 NppStatus nppiCopyReplicateBorder\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 32-bit floating point image copy with nearest source image pixel color.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region of pixels.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).  
*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .  
*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.11 NppStatus nppiCopyReplicateBorder\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

3 channel 32-bit floating point image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.12 NppStatus nppiCopyReplicateBorder\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

4 channel 32-bit floating point image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.13** `NppStatus nppiCopyReplicateBorder_32s_AC4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 32-bit signed integer image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.14** `NppStatus nppiCopyReplicateBorder_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

1 channel 32-bit signed integer image copy with nearest source image pixel color.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region of pixels.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).  
*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .  
*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.15** `NppStatus nppiCopyReplicateBorder_32s_C3R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

3 channel 32-bit signed image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.16** `NppStatus nppiCopyReplicateBorder_32s_C4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 32-bit signed integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.17** `NppStatus nppiCopyReplicateBorder_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 8-bit unsigned integer image copy with nearest source image pixel color with alpha channel unaffected.

See [nppiCopyReplicateBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.18** `NppStatus nppiCopyReplicateBorder_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

1 channel 8-bit unsigned integer image copy with nearest source image pixel color.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region of pixels.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and nearest source image pixel color (outer part).  
*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the nearest source image pixel color.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .  
*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.19 NppStatus nppiCopyReplicateBorder\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

3 channel 8-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.59.1.20 NppStatus nppiCopyReplicateBorder\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

4 channel 8-bit unsigned integer image copy with nearest source image pixel color.

See [nppiCopyReplicateBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



## 7.60 Copy Wrap Border

### CopyWrapBorder

Methods for copying images and padding borders with wrapped replications of the source image pixel colors.

- **NppStatus nppiCopyWrapBorder\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*1 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.*
- **NppStatus nppiCopyWrapBorder\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*3 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.*
- **NppStatus nppiCopyWrapBorder\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.*
- **NppStatus nppiCopyWrapBorder\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.*
- **NppStatus nppiCopyWrapBorder\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*1 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.*
- **NppStatus nppiCopyWrapBorder\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*3 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.*
- **NppStatus nppiCopyWrapBorder\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.*
- **NppStatus nppiCopyWrapBorder\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)  
*4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.*

*4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.*

- **NppStatus nppiCopyWrapBorder\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*1 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*3 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.*

- **NppStatus nppiCopyWrapBorder\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*1 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_32s\_C3R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*3 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_32s\_C4R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_32s\_AC4R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.*

- **NppStatus nppiCopyWrapBorder\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*3 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*4 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.*

- **NppStatus nppiCopyWrapBorder\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSrcSizeROI, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

*1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.*

## 7.60.1 Function Documentation

### 7.60.1.1 NppStatus nppiCopyWrapBorder\_16s\_AC4R (const Npp16s \*pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp16s \*pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**oSrcSizeROI** Size of the source region-of-interest.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oDstSizeROI** Size of the destination region-of-interest.  
**nTopBorderHeight** Height of top border.  
**nLeftBorderWidth** Width of left border.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.60.1.2 NppStatus nppiCopyWrapBorder\_16s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.60.1.3 NppStatus nppiCopyWrapBorder\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.4 NppStatus nppiCopyWrapBorder\_16s\_C4R** (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 16-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_16s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.5 NppStatus nppiCopyWrapBorder\_16u\_AC4R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.60.1.6 **NppStatus nppiCopyWrapBorder\_16u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

1 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.60.1.7 **NppStatus nppiCopyWrapBorder\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)

3 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.8 NppStatus nppiCopyWrapBorder\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

4 channel 16-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.9 NppStatus nppiCopyWrapBorder\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.10 NppStatus nppiCopyWrapBorder\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

1 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.11 NppStatus nppiCopyWrapBorder\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

3 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.60.1.12 NppStatus nppiCopyWrapBorder\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

4 channel 32-bit floating point image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_32f\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.13 NppStatus nppiCopyWrapBorder\_32s\_AC4R (const Npp32s \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.14 NppStatus nppiCopyWrapBorder\_32s\_C1R (const Npp32s \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

1 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.15 NppStatus nppiCopyWrapBorder\_32s\_C3R (const Npp32s \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

3 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.16** `NppStatus nppiCopyWrapBorder_32s_C4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 32-bit signed integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.17** `NppStatus nppiCopyWrapBorder_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors with alpha channel unaffected.

See [nppiCopyWrapBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSizeROI* Size of the source region-of-interest.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nTopBorderHeight* Height of top border.  
*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.18 NppStatus nppiCopyWrapBorder\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

1 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region of pixels.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image (inner part) and a border consisting of wrapped replication of the source image pixel colors (outer part).

*nTopBorderHeight* Height (in pixels) of the top border. The number of pixel rows at the top of the destination ROI that will be filled with the wrapped replication of the corresponding column of source image pixels colors.  $nBottomBorderHeight = oDstSizeROI.height - nTopBorderHeight - oSrcSizeROI.height$ .

*nLeftBorderWidth* Width (in pixels) of the left border. The width of the border at the right side of the destination ROI is implicitly defined by the size of the source ROI:  $nRightBorderWidth = oDstSizeROI.width - nLeftBorderWidth - oSrcSizeROI.width$ .

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.19 NppStatus nppiCopyWrapBorder\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSrcSizeROI*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *nTopBorderHeight*, int *nLeftBorderWidth*)**

3 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.60.1.20** `NppStatus nppiCopyWrapBorder_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSizeROI, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int nTopBorderHeight, int nLeftBorderWidth)`

4 channel 8-bit unsigned integer image copy with the borders wrapped by replication of source image pixel colors.

See [nppiCopyWrapBorder\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSizeROI* Size of the source region-of-interest.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nTopBorderHeight* Height of top border.

*nLeftBorderWidth* Width of left border.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.61 Copy Sub-Pixel

### CopySubpix

Functions for copying linearly interpolated images using source image subpixel coordinates

- **NppStatus nppiCopySubpix\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*1 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*3 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.*
- **NppStatus nppiCopySubpix\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*1 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*3 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 16-bit unsigned linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.*
- **NppStatus nppiCopySubpix\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*1 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*3 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.*

- **NppStatus nppiCopySubpix\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.*
- **NppStatus nppiCopySubpix\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*1 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_32s\_C3R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*3 channel 32-bit signed linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_32s\_C4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_32s\_AC4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.*
- **NppStatus nppiCopySubpix\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*1 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*3 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.*
- **NppStatus nppiCopySubpix\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)  
*4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.*

### 7.61.1 Function Documentation

#### 7.61.1.1 **NppStatus nppiCopySubpix\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, **Npp32f** nDx, **Npp32f** nDy)

4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix\\_16s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.2 **NppStatus nppiCopySubpix\_16s\_C1R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.3 **NppStatus nppiCopySubpix\_16s\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_16s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Size of the destination region-of-interest.



*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.61.1.4 NppStatus nppiCopySubpix\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)**

4 channel 16-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_16s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.61.1.5 NppStatus nppiCopySubpix\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)**

4 channel 16-bit unsigned linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.61.1.6 NppStatus nppiCopySubpix\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)**

1 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.61.1.7 NppStatus nppiCopySubpix\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)**

3 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.61.1.8 NppStatus nppiCopySubpix\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)**

4 channel 16-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_16u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.9 **NppStatus nppiCopySubpix\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix\\_32f\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.10 **NppStatus nppiCopySubpix\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.11 **NppStatus nppiCopySubpix\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_32f\\_C1R\(\)](#) for detailed documentation.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.12 **NppStatus nppiCopySubpix\_32f\_C4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 32-bit floating point linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_32f\\_C1R\(\)](#) for detailed documentation.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.13 **NppStatus nppiCopySubpix\_32s\_AC4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.14 **NppStatus nppiCopySubpix\_32s\_C1R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

1 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.  
*nDx* Fractional part of source image X coordinate.  
*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.61.1.15 **NppStatus nppiCopySubpix\_32s\_C3R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, Npp32f *nDx*, Npp32f *nDy*)

3 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.61.1.16 NppStatus nppiCopySubpix\_32s\_C4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)**

4 channel 32-bit signed integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_32s\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.61.1.17 NppStatus nppiCopySubpix\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)**

4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy with alpha channel unaffected.

See [nppiCopySubpix\\_8u\\_C1R\(\)](#) for detailed documentation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.61.1.18 `NppStatus nppiCopySubpix_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)`

1 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.61.1.19 `NppStatus nppiCopySubpix_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)`

3 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_8u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.61.1.20 `NppStatus nppiCopySubpix_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, Npp32f nDx, Npp32f nDy)`

4 channel 8-bit unsigned integer linearly interpolated source image subpixel coordinate color copy.

See [nppiCopySubpix\\_8u\\_C1R\(\)](#) for detailed documentation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Size of the destination region-of-interest.

*nDx* Fractional part of source image X coordinate.

*nDy* Fractional part of source image Y coordinate.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



## 7.62 Duplicate Channel

### Dup

Functions for duplicating a single channel image in a multiple channel image.

- **NppStatus nppiDup\_8u\_C1C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 8-bit unsigned integer source image duplicated in all 3 channels of destination image.*
- **NppStatus nppiDup\_8u\_C1C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 8-bit unsigned integer source image duplicated in all 4 channels of destination image.*
- **NppStatus nppiDup\_8u\_C1AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 8-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.*
- **NppStatus nppiDup\_16u\_C1C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 16-bit unsigned integer source image duplicated in all 3 channels of destination image.*
- **NppStatus nppiDup\_16u\_C1C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 16-bit unsigned integer source image duplicated in all 4 channels of destination image.*
- **NppStatus nppiDup\_16u\_C1AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 16-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.*
- **NppStatus nppiDup\_16s\_C1C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 16-bit signed integer source image duplicated in all 3 channels of destination image.*
- **NppStatus nppiDup\_16s\_C1C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 16-bit signed integer source image duplicated in all 4 channels of destination image.*
- **NppStatus nppiDup\_16s\_C1AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 16-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.*
- **NppStatus nppiDup\_32s\_C1C3R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)  
*1 channel 32-bit signed integer source image duplicated in all 3 channels of destination image.*
- **NppStatus nppiDup\_32s\_C1C4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)

*1 channel 32-bit signed integer source image duplicated in all 4 channels of destination image.*

- **NppStatus nppiDup\_32s\_C1AC4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)

*1 channel 32-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.*

- **NppStatus nppiDup\_32f\_C1C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)

*1 channel 32-bit floating point source image duplicated in all 3 channels of destination image.*

- **NppStatus nppiDup\_32f\_C1C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)

*1 channel 32-bit floating point source image duplicated in all 4 channels of destination image.*

- **NppStatus nppiDup\_32f\_C1AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)

*1 channel 32-bit floating point source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.*

## 7.62.1 Function Documentation

### 7.62.1.1 **NppStatus nppiDup\_16s\_C1AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 16-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oDstSizeROI** Size of the destination region-of-interest.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.62.1.2 **NppStatus nppiDup\_16s\_C1C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI)

1 channel 16-bit signed integer source image duplicated in all 3 channels of destination image.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.62.1.3 NppStatus nppiDup\_16s\_C1C4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oDstSizeROI)**

1 channel 16-bit signed integer source image duplicated in all 4 channels of destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.62.1.4 NppStatus nppiDup\_16u\_C1AC4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oDstSizeROI)**

1 channel 16-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.62.1.5 NppStatus nppiDup\_16u\_C1C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 16-bit unsigned integer source image duplicated in all 3 channels of destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.6 NppStatus nppiDup\_16u\_C1C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 16-bit unsigned integer source image duplicated in all 4 channels of destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.7 NppStatus nppiDup\_32f\_C1AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 32-bit floating point source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.8 NppStatus nppiDup\_32f\_C1C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 32-bit floating point source image duplicated in all 3 channels of destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.9 NppStatus nppiDup\_32f\_C1C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 32-bit floating point source image duplicated in all 4 channels of destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.10 NppStatus nppiDup\_32s\_C1AC4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)

1 channel 32-bit signed integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.11 `NppStatus nppiDup_32s_C1C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI)`

1 channel 32-bit signed integer source image duplicated in all 3 channels of destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.12 `NppStatus nppiDup_32s_C1C4R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oDstSizeROI)`

1 channel 32-bit signed integer source image duplicated in all 4 channels of destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.62.1.13 `NppStatus nppiDup_8u_C1AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI)`

1 channel 8-bit unsigned integer source image duplicated in 3 channels of 4 channel destination image with alpha channel unaffected.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.62.1.14 NppStatus nppiDup\_8u\_C1C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)**

1 channel 8-bit unsigned integer source image duplicated in all 3 channels of destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size (width, height) of the destination region, i.e. the region that gets filled with data from the source image, source image ROI is assumed to be same as destination image ROI.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.62.1.15 NppStatus nppiDup\_8u\_C1C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*)**

1 channel 8-bit unsigned integer source image duplicated in all 4 channels of destination image.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstSizeROI* Size of the destination region-of-interest.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.63 Transpose

### Transpose

Methods for transposing images of various types.

Like matrix transpose, image transpose is a mirror along the image's diagonal (upper-left to lower-right corner).

- `NppStatus nppiTranspose_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*1 channel 8-bit unsigned int image transpose.*
- `NppStatus nppiTranspose_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*3 channel 8-bit unsigned int image transpose.*
- `NppStatus nppiTranspose_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*4 channel 8-bit unsigned int image transpose.*
- `NppStatus nppiTranspose_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*1 channel 16-bit unsigned int image transpose.*
- `NppStatus nppiTranspose_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*3 channel 16-bit unsigned int image transpose.*
- `NppStatus nppiTranspose_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*4 channel 16-bit unsigned int image transpose.*
- `NppStatus nppiTranspose_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*1 channel 16-bit signed int image transpose.*
- `NppStatus nppiTranspose_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*3 channel 16-bit signed int image transpose.*
- `NppStatus nppiTranspose_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*4 channel 16-bit signed int image transpose.*
- `NppStatus nppiTranspose_32s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSrcROI)  
*1 channel 32-bit signed int image transpose.*
- `NppStatus nppiTranspose_32s_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSrcROI)



3 channel 32-bit signed int image transpose.

- **NppStatus nppiTranspose\_32s\_C4R** (const **Npp32s** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSrcROI)

4 channel 32-bit signed int image transpose.

- **NppStatus nppiTranspose\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSrcROI)

1 channel 32-bit floating point image transpose.

- **NppStatus nppiTranspose\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSrcROI)

3 channel 32-bit floating point image transpose.

- **NppStatus nppiTranspose\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSrcROI)

4 channel 32-bit floating point image transpose.

## 7.63.1 Function Documentation

### 7.63.1.1 **NppStatus nppiTranspose\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSrcROI)

1 channel 16-bit signed int image transpose.

#### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pDst** Pointer to the destination ROI.

**nDstStep** Destination-Image Line Step.

**oSrcROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.63.1.2 **NppStatus nppiTranspose\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSrcROI)

3 channel 16-bit signed int image transpose.

#### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pDst** Pointer to the destination ROI.

**nDstStep** Destination-Image Line Step.

*oSrcROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.63.1.3 NppStatus nppiTranspose\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSrcROI)**

4 channel 16-bit signed int image transpose.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Pointer to the destination ROI](#).

*nDstStep* [Destination-Image Line Step](#).

*oSrcROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.63.1.4 NppStatus nppiTranspose\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSrcROI)**

1 channel 16-bit unsigned int image transpose.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Pointer to the destination ROI](#).

*nDstStep* [Destination-Image Line Step](#).

*oSrcROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.63.1.5 NppStatus nppiTranspose\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSrcROI)**

3 channel 16-bit unsigned int image transpose.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* Pointer to the destination ROI.  
*nDstStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.63.1.6 NppStatus nppiTranspose\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

4 channel 16-bit unsigned int image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Pointer to the destination ROI.  
*nDstStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.63.1.7 NppStatus nppiTranspose\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

1 channel 32-bit floating point image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Pointer to the destination ROI.  
*nDstStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.63.1.8 NppStatus nppiTranspose\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)

3 channel 32-bit floating point image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Pointer to the destination ROI.

*nDstStep* Destination-Image Line Step.

*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.63.1.9 NppStatus nppiTranspose\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSrcROI)**

4 channel 32-bit floating point image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Pointer to the destination ROI.

*nDstStep* Destination-Image Line Step.

*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.63.1.10 NppStatus nppiTranspose\_32s\_C1R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSrcROI)**

1 channel 32-bit signed int image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Pointer to the destination ROI.

*nDstStep* Destination-Image Line Step.

*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.63.1.11 NppStatus nppiTranspose\_32s\_C3R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)**

3 channel 32-bit signed int image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Pointer to the destination ROI.  
*nDstStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.63.1.12 NppStatus nppiTranspose\_32s\_C4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)**

4 channel 32-bit signed int image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Pointer to the destination ROI.  
*nDstStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.63.1.13 NppStatus nppiTranspose\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)**

1 channel 8-bit unsigned int image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Pointer to the destination ROI.  
*nDstStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.63.1.14 NppStatus nppiTranspose\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)**

3 channel 8-bit unsigned int image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Pointer to the destination ROI.

*nDstStep* Destination-Image Line Step.

*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.63.1.15 NppStatus nppiTranspose\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSrcROI*)**

4 channel 8-bit unsigned int image transpose.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Pointer to the destination ROI.

*nDstStep* Destination-Image Line Step.

*oSrcROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.64 Swap Channels

### SwapChannels

Functions for swapping and duplicating channels in multiple channel images.

The methods support arbitrary permutations of the original channels, including replication and setting one or more channels to a constant value.

- **NppStatus** **nppiSwapChannels\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])  
*3 channel 8-bit unsigned integer source image to 3 channel destination image.*
- **NppStatus** **nppiSwapChannels\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])  
*3 channel 8-bit unsigned integer in place image.*
- **NppStatus** **nppiSwapChannels\_8u\_C4C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])  
*4 channel 8-bit unsigned integer source image to 3 channel destination image.*
- **NppStatus** **nppiSwapChannels\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[4])  
*4 channel 8-bit unsigned integer source image to 4 channel destination image.*
- **NppStatus** **nppiSwapChannels\_8u\_C4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const int aDstOrder[4])  
*4 channel 8-bit unsigned integer in place image.*
- **NppStatus** **nppiSwapChannels\_8u\_C3C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[4], const **Npp8u** nValue)  
*3 channel 8-bit unsigned integer source image to 4 channel destination image.*
- **NppStatus** **nppiSwapChannels\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])  
*4 channel 8-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.*
- **NppStatus** **nppiSwapChannels\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])  
*3 channel 16-bit unsigned integer source image to 3 channel destination image.*
- **NppStatus** **nppiSwapChannels\_16u\_C3IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])  
*3 channel 16-bit unsigned integer in place image.*
- **NppStatus** **nppiSwapChannels\_16u\_C4C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[3])  
*4 channel 16-bit unsigned integer source image to 3 channel destination image.*
- **NppStatus** **nppiSwapChannels\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const int aDstOrder[4])

*4 channel 16-bit unsigned integer source image to 4 channel destination image.*

- `NppStatus nppiSwapChannels_16u_C4IR (Npp16u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

*4 channel 16-bit unsigned integer in place image.*

- `NppStatus nppiSwapChannels_16u_C3C4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp16u nValue)`

*3 channel 16-bit unsigned integer source image to 4 channel destination image.*

- `NppStatus nppiSwapChannels_16u_AC4R (const Npp16u *pSrc, int nSrcStep, Npp16u *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

*4 channel 16-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.*

- `NppStatus nppiSwapChannels_16s_C3R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

*3 channel 16-bit signed integer source image to 3 channel destination image.*

- `NppStatus nppiSwapChannels_16s_C3IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

*3 channel 16-bit signed integer in place image.*

- `NppStatus nppiSwapChannels_16s_C4C3R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

*4 channel 16-bit signed integer source image to 3 channel destination image.*

- `NppStatus nppiSwapChannels_16s_C4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

*4 channel 16-bit signed integer source image to 4 channel destination image.*

- `NppStatus nppiSwapChannels_16s_C4IR (Npp16s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

*4 channel 16-bit signed integer in place image.*

- `NppStatus nppiSwapChannels_16s_C3C4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp16s nValue)`

*3 channel 16-bit signed integer source image to 4 channel destination image.*

- `NppStatus nppiSwapChannels_16s_AC4R (const Npp16s *pSrc, int nSrcStep, Npp16s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

*4 channel 16-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.*

- `NppStatus nppiSwapChannels_32s_C3R (const Npp32s *pSrc, int nSrcStep, Npp32s *pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

*3 channel 32-bit signed integer source image to 3 channel destination image.*

- `NppStatus nppiSwapChannels_32s_C3IR (Npp32s *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

*3 channel 32-bit signed integer in place image.*



- `NppStatus nppiSwapChannels_32s_C4C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[3])  
*4 channel 32-bit signed integer source image to 3 channel destination image.*
- `NppStatus nppiSwapChannels_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[4])  
*4 channel 32-bit signed integer source image to 4 channel destination image.*
- `NppStatus nppiSwapChannels_32s_C4IR` (`Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const int aDstOrder[4])  
*4 channel 32-bit signed integer in place image.*
- `NppStatus nppiSwapChannels_32s_C3C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[4], const `Npp32s` nValue)  
*3 channel 32-bit signed integer source image to 4 channel destination image.*
- `NppStatus nppiSwapChannels_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[3])  
*4 channel 32-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.*
- `NppStatus nppiSwapChannels_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[3])  
*3 channel 32-bit floating point source image to 3 channel destination image.*
- `NppStatus nppiSwapChannels_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const int aDstOrder[3])  
*3 channel 32-bit floating point in place image.*
- `NppStatus nppiSwapChannels_32f_C4C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[3])  
*4 channel 32-bit floating point source image to 3 channel destination image.*
- `NppStatus nppiSwapChannels_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[4])  
*4 channel 32-bit floating point source image to 4 channel destination image.*
- `NppStatus nppiSwapChannels_32f_C4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const int aDstOrder[4])  
*4 channel 32-bit floating point in place image.*
- `NppStatus nppiSwapChannels_32f_C3C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[4], const `Npp32f` nValue)  
*3 channel 32-bit floating point source image to 4 channel destination image.*
- `NppStatus nppiSwapChannels_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const int aDstOrder[3])  
*4 channel 32-bit floating point source image to 4 channel destination image with destination alpha channel unaffected.*

## 7.64.1 Function Documentation

### 7.64.1.1 `NppStatus nppiSwapChannels_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 16-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.2 `NppStatus nppiSwapChannels_16s_C3C4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp16s nValue)`

3 channel 16-bit signed integer source image to 4 channel destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

*nValue* (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.3 NppStatus nppiSwapChannels\_16s\_C3IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 16-bit signed integer in place image.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.4 NppStatus nppiSwapChannels\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 16-bit signed integer source image to 3 channel destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.5 NppStatus nppiSwapChannels\_16s\_C4C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 16-bit signed integer source image to 3 channel destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.6 NppStatus nppiSwapChannels\_16s\_C4IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4])**

4 channel 16-bit signed integer in place image.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.7 NppStatus nppiSwapChannels\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4])**

4 channel 16-bit signed integer source image to 4 channel destination image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.8 NppStatus nppiSwapChannels\_16u\_AC4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 16-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.9 NppStatus nppiSwapChannels\_16u\_C3C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4], const Npp16u *nValue*)

3 channel 16-bit unsigned integer source image to 4 channel destination image.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

*nValue* (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.10 NppStatus nppiSwapChannels\_16u\_C3IR (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 16-bit unsigned integer in place image.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.11 NppStatus nppiSwapChannels\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 16-bit unsigned integer source image to 3 channel destination image.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.12 NppStatus nppiSwapChannels\_16u\_C4C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 16-bit unsigned integer source image to 3 channel destination image.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.13 NppStatus nppiSwapChannels\_16u\_C4IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])**

4 channel 16-bit unsigned integer in place image.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.14 NppStatus nppiSwapChannels\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])**

4 channel 16-bit unsigned integer source image to 4 channel destination image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.15 `NppStatus nppiSwapChannels_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 32-bit floating point source image to 4 channel destination image with destination alpha channel unaffected.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, `aDstOrder = [2,1,0]` converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.16 `NppStatus nppiSwapChannels_32f_C3C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp32f nValue)`

3 channel 32-bit floating point source image to 4 channel destination image.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, `aDstOrder = [3,2,1,0]` converts this to VBGR channel order.

*nValue* (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. `nValue` is either written or not written to a particular channel depending on the `aDstOrder` entry for that destination channel. An `aDstOrder` value of 3 will output `nValue` to that channel, an `aDstOrder` value greater than 3 will leave that particular destination channel value unmodified.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.64.1.17 `NppStatus nppiSwapChannels_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 32-bit floating point in place image.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [oSizeROI Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.18 `NppStatus nppiSwapChannels_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

3 channel 32-bit floating point source image to 3 channel destination image.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.19 `NppStatus nppiSwapChannels_32f_C4C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 32-bit floating point source image to 3 channel destination image.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.20** `NppStatus nppiSwapChannels_32f_C4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 32-bit floating point in place image.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.21** `NppStatus nppiSwapChannels_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 32-bit floating point source image to 4 channel destination image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.22 NppStatus nppiSwapChannels\_32s\_AC4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 32-bit signed integer source image to 4 channel destination image with destination alpha channel unaffected.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.64.1.23 NppStatus nppiSwapChannels\_32s\_C3C4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[4], const Npp32s *nValue*)

3 channel 32-bit signed integer source image to 4 channel destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

*nValue* (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.64.1.24 **NppStatus nppiSwapChannels\_32s\_C3IR** (Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 32-bit signed integer in place image.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.25 **NppStatus nppiSwapChannels\_32s\_C3R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 32-bit signed integer source image to 3 channel destination image.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.64.1.26 **NppStatus nppiSwapChannels\_32s\_C4C3R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 32-bit signed integer source image to 3 channel destination image.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.27 NppStatus nppiSwapChannels\_32s\_C4IR (Npp32s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])**

4 channel 32-bit signed integer in place image.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.28 NppStatus nppiSwapChannels\_32s\_C4R (const Npp32s \* pSrc, int nSrcStep, Npp32s \* pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])**

4 channel 32-bit signed integer source image to 4 channel destination image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.29 `NppStatus nppiSwapChannels_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[3])`

4 channel 8-bit unsigned integer source image to 4 channel destination image with destination alpha channel unaffected.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order. of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to BGRA channel order. In the AC4R case, the alpha channel is always assumed to be channel 3.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.30 `NppStatus nppiSwapChannels_8u_C3C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4], const Npp8u nValue)`

3 channel 8-bit unsigned integer source image to 4 channel destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [3,2,1,0] converts this to VBGR channel order.

*nValue* (V) Single channel constant value that can be replicated in one or more of the 4 destination channels. *nValue* is either written or not written to a particular channel depending on the *aDstOrder* entry for that destination channel. An *aDstOrder* value of 3 will output *nValue* to that channel, an *aDstOrder* value greater than 3 will leave that particular destination channel value unmodified.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.31 NppStatus nppiSwapChannels\_8u\_C3IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 8-bit unsigned integer in place image.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.32 NppStatus nppiSwapChannels\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

3 channel 8-bit unsigned integer source image to 3 channel destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGB image, *aDstOrder* = [2,1,0] converts this to BGR channel order.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.64.1.33 NppStatus nppiSwapChannels\_8u\_C4C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const int *aDstOrder*[3])

4 channel 8-bit unsigned integer source image to 3 channel destination image.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an RGBA image, *aDstOrder* = [2,1,0] converts this to a 3 channel BGR channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.34** `NppStatus nppiSwapChannels_8u_C4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 8-bit unsigned integer in place image.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.64.1.35** `NppStatus nppiSwapChannels_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const int aDstOrder[4])`

4 channel 8-bit unsigned integer source image to 4 channel destination image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDstOrder* Host memory integer array describing how channel values are permuted. The n-th entry of the array contains the number of the channel that is stored in the n-th channel of the output image. E.g. Given an ARGB image, *aDstOrder* = [3,2,1,0] converts this to BGRA channel order.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



## 7.65 Filtering Functions

Linear and non-linear image filtering functions.

### Modules

- [1D Linear Filter](#)

### FilterSobelVertSecondBorder

Filters the image using a second derivative, vertical Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 & 0 & 1 \\ 4 & 0 & -8 & 0 & 4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & 0 & -8 & 0 & 4 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelVertSecondBorder_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertSecondBorder_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertSecondBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 32-bit floating-point second derivative, vertical Sobel filter with border control.*

### FilterSobelCrossBorder

Filters the image using a second cross derivative Sobel filter kernel with border control:

$$\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & -4 & -2 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelCrossBorder_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter with border control.*

- `NppStatus nppiFilterSobelCrossBorder_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter with border control.*

- `NppStatus nppiFilterSobelCrossBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 32-bit floating-point second cross derivative Sobel filter with border control.*

## FilterRobertsDown

Filters the image using a horizontal Roberts filter kernel:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterRobertsDown_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit unsigned horizontal Roberts filter.*

- `NppStatus nppiFilterRobertsDown_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit unsigned horizontal Roberts filter.*

- `NppStatus nppiFilterRobertsDown_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned horizontal Roberts filter.*

- `NppStatus nppiFilterRobertsDown_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned horizontal Roberts filter, ignoring alpha-channel.*

- `NppStatus nppiFilterRobertsDown_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed horizontal Roberts filter.*

- `NppStatus nppiFilterRobertsDown_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit signed horizontal Roberts filter.*

- `NppStatus nppiFilterRobertsDown_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed horizontal Roberts filter.*

- `NppStatus nppiFilterRobertsDown_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed horizontal Roberts filter, ignoring alpha-channel.*

- `NppStatus nppiFilterRobertsDown_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit floating-point horizontal Roberts filter.*
- `NppStatus nppiFilterRobertsDown_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit floating-point horizontal Roberts filter.*
- `NppStatus nppiFilterRobertsDown_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point horizontal Roberts filter.*
- `NppStatus nppiFilterRobertsDown_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point horizontal Roberts filter; ignoring alpha-channel.*

## FilterRobertsUp

Filters the image using a vertical Roberts filter kernel:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterRobertsUp_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit unsigned vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 8-bit unsigned vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned vertical Roberts filter; ignoring alpha-channel.*
- `NppStatus nppiFilterRobertsUp_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit signed vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit signed vertical Roberts filter.*

- `NppStatus nppiFilterRobertsUp_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed vertical Roberts filter; ignoring alpha-channel.*
- `NppStatus nppiFilterRobertsUp_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit floating-point vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit floating-point vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point vertical Roberts filter.*
- `NppStatus nppiFilterRobertsUp_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point vertical Roberts filter; ignoring alpha-channel.*

## FilterLaplace

Filters the image using a Laplacian filter kernel:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -3 & -4 & -3 & -1 \\ -3 & 0 & 6 & 0 & -3 \\ -4 & 6 & 20 & 6 & -4 \\ -3 & 0 & 6 & 0 & -3 \\ -1 & -3 & -4 & -3 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterLaplace_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit unsigned Laplace filter.*
- `NppStatus nppiFilterLaplace_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Three channel 8-bit unsigned Laplace filter.*
- `NppStatus nppiFilterLaplace_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 8-bit unsigned Laplace filter.*
- `NppStatus nppiFilterLaplace_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 8-bit unsigned Laplace filter; ignoring alpha channel.*

- `NppStatus nppiFilterLaplace_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 16-bit signed Laplace filter.*
- `NppStatus nppiFilterLaplace_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Three channel 16-bit signed Laplace filter.*
- `NppStatus nppiFilterLaplace_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 16-bit signed Laplace filter.*
- `NppStatus nppiFilterLaplace_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 16-bit signed Laplace filter, ignoring alpha channel.*
- `NppStatus nppiFilterLaplace_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 32-bit floating-point Laplace filter.*
- `NppStatus nppiFilterLaplace_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Three channel 32-bit floating-point Laplace filter.*
- `NppStatus nppiFilterLaplace_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 32-bit floating-point Laplace filter.*
- `NppStatus nppiFilterLaplace_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 32-bit floating-point Laplace filter, ignoring alpha channel.*
- `NppStatus nppiFilterLaplace_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit unsigned to 16-bit signed Laplace filter.*
- `NppStatus nppiFilterLaplace_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit signed to 16-bit signed Laplace filter.*

## FilterGauss

Filters the image using a Gaussian filter kernel:

$$\begin{pmatrix} 1/16 & 2/16 & 1/16 \\ 2/16 & 4/16 & 2/16 \\ 1/16 & 2/16 & 1/16 \end{pmatrix} \begin{pmatrix} 2/571 & 7/571 & 12/571 & 7/571 & 2/571 \\ 7/571 & 31/571 & 52/571 & 31/571 & 7/571 \\ 12/571 & 52/571 & 127/571 & 52/571 & 12/571 \\ 7/571 & 31/571 & 52/571 & 31/571 & 7/571 \\ 2/571 & 7/571 & 12/571 & 7/571 & 2/571 \end{pmatrix}$$

- `NppStatus nppiFilterGauss_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit unsigned Gauss filter.*
- `NppStatus nppiFilterGauss_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Three channel 8-bit unsigned Gauss filter.*
- `NppStatus nppiFilterGauss_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 8-bit unsigned Gauss filter.*
- `NppStatus nppiFilterGauss_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.*
- `NppStatus nppiFilterGauss_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 16-bit unsigned Gauss filter.*
- `NppStatus nppiFilterGauss_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Three channel 16-bit unsigned Gauss filter.*
- `NppStatus nppiFilterGauss_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 16-bit unsigned Gauss filter.*
- `NppStatus nppiFilterGauss_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.*
- `NppStatus nppiFilterGauss_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 16-bit signed Gauss filter.*
- `NppStatus nppiFilterGauss_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Three channel 16-bit signed Gauss filter.*
- `NppStatus nppiFilterGauss_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 16-bit signed Gauss filter.*
- `NppStatus nppiFilterGauss_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Four channel 16-bit signed Gauss filter, ignoring alpha channel.*
- `NppStatus nppiFilterGauss_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 32-bit floating-point Gauss filter.*

- `NppStatus nppiFilterGauss_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 32-bit floating-point Gauss filter.*

- `NppStatus nppiFilterGauss_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 32-bit floating-point Gauss filter.*

- `NppStatus nppiFilterGauss_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.*

## FilterHighPass

Filters the image using a high-pass filter kernel:

$$\begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterHighPass_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 8-bit unsigned high-pass filter.*

- `NppStatus nppiFilterHighPass_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 8-bit unsigned high-pass filter.*

- `NppStatus nppiFilterHighPass_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 8-bit unsigned high-pass filter.*

- `NppStatus nppiFilterHighPass_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.*

- `NppStatus nppiFilterHighPass_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 16-bit unsigned high-pass filter.*

- `NppStatus nppiFilterHighPass_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 16-bit unsigned high-pass filter.*

- `NppStatus nppiFilterHighPass_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit unsigned high-pass filter.*

- `NppStatus nppiFilterHighPass_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.*

- `NppStatus nppiFilterHighPass_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 16-bit signed high-pass filter.*

- `NppStatus nppiFilterHighPass_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 16-bit signed high-pass filter.*

- `NppStatus nppiFilterHighPass_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit signed high-pass filter.*

- `NppStatus nppiFilterHighPass_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit signed high-pass filter, ignoring alpha channel.*

- `NppStatus nppiFilterHighPass_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 32-bit floating-point high-pass filter.*

- `NppStatus nppiFilterHighPass_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 32-bit floating-point high-pass filter.*

- `NppStatus nppiFilterHighPass_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 32-bit floating-point high-pass filter.*

- `NppStatus nppiFilterHighPass_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.*

## FilterLowPass

Filters the image using a low-pass filter kernel:

$$\begin{pmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{pmatrix} \begin{pmatrix} 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \\ 1/25 & 1/25 & 1/25 & 1/25 & 1/25 \end{pmatrix}$$

- `NppStatus nppiFilterLowPass_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)



*Single channel 8-bit unsigned low-pass filter.*

- `NppStatus nppiFilterLowPass_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 8-bit unsigned low-pass filter.*

- `NppStatus nppiFilterLowPass_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 8-bit unsigned low-pass filter.*

- `NppStatus nppiFilterLowPass_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 8-bit unsigned low-pass filter, ignoring alpha channel.*

- `NppStatus nppiFilterLowPass_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 16-bit unsigned low-pass filter.*

- `NppStatus nppiFilterLowPass_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 16-bit unsigned low-pass filter.*

- `NppStatus nppiFilterLowPass_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit unsigned low-pass filter.*

- `NppStatus nppiFilterLowPass_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit unsigned low-pass filter, ignoring alpha channel.*

- `NppStatus nppiFilterLowPass_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 16-bit signed low-pass filter.*

- `NppStatus nppiFilterLowPass_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Three channel 16-bit signed low-pass filter.*

- `NppStatus nppiFilterLowPass_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit signed low-pass filter.*

- `NppStatus nppiFilterLowPass_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Four channel 16-bit signed low-pass filter, ignoring alpha channel.*

- `NppStatus nppiFilterLowPass_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 32-bit floating-point low-pass filter.*

- `NppStatus nppiFilterLowPass_32f_C3R` (const `Npp32f *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`)  
*Three channel 32-bit floating-point low-pass filter.*
- `NppStatus nppiFilterLowPass_32f_C4R` (const `Npp32f *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`)  
*Four channel 32-bit floating-point low-pass filter.*
- `NppStatus nppiFilterLowPass_32f_AC4R` (const `Npp32f *pSrc`, `Npp32s nSrcStep`, `Npp32f *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`, `NppiMaskSize eMaskSize`)  
*Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.*

## FilterSharpen

Filters the image using a sharpening filter kernel:

$$\begin{pmatrix} -1/8 & -1/8 & -1/8 \\ -1/8 & 16/8 & -1/8 \\ -1/8 & -1/8 & -1/8 \end{pmatrix}$$

- `NppStatus nppiFilterSharpen_8u_C1R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)  
*Single channel 8-bit unsigned sharpening filter.*
- `NppStatus nppiFilterSharpen_8u_C3R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)  
*Three channel 8-bit unsigned sharpening filter.*
- `NppStatus nppiFilterSharpen_8u_C4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)  
*Four channel 8-bit unsigned sharpening filter.*
- `NppStatus nppiFilterSharpen_8u_AC4R` (const `Npp8u *pSrc`, `Npp32s nSrcStep`, `Npp8u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)  
*Four channel 8-bit unsigned sharpening filter; ignoring alpha channel.*
- `NppStatus nppiFilterSharpen_16u_C1R` (const `Npp16u *pSrc`, `Npp32s nSrcStep`, `Npp16u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)  
*Single channel 16-bit unsigned sharpening filter.*
- `NppStatus nppiFilterSharpen_16u_C3R` (const `Npp16u *pSrc`, `Npp32s nSrcStep`, `Npp16u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)  
*Three channel 16-bit unsigned sharpening filter.*
- `NppStatus nppiFilterSharpen_16u_C4R` (const `Npp16u *pSrc`, `Npp32s nSrcStep`, `Npp16u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)  
*Four channel 16-bit unsigned sharpening filter.*
- `NppStatus nppiFilterSharpen_16u_AC4R` (const `Npp16u *pSrc`, `Npp32s nSrcStep`, `Npp16u *pDst`, `Npp32s nDstStep`, `NppiSize oSizeROI`)

*Four channel 16-bit unsigned sharpening filter, ignoring alpha channel.*

- `NppStatus nppiFilterSharpen_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed sharpening filter.*

- `NppStatus nppiFilterSharpen_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit signed sharpening filter.*

- `NppStatus nppiFilterSharpen_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed sharpening filter.*

- `NppStatus nppiFilterSharpen_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed sharpening filter, ignoring alpha channel.*

- `NppStatus nppiFilterSharpen_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 32-bit floating-point sharpening filter.*

- `NppStatus nppiFilterSharpen_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 32-bit floating-point sharpening filter.*

- `NppStatus nppiFilterSharpen_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 32-bit floating-point sharpening filter.*

- `NppStatus nppiFilterSharpen_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 32-bit floating-point sharpening filter, ignoring alpha channel.*

### 7.65.1 Detailed Description

Linear and non-linear image filtering functions.

Filtering functions are classified as [Neighborhood Operations](#). It is the user's responsibility to avoid [Sampling Beyond Image Boundaries](#).

### 7.65.2 Function Documentation

#### 7.65.2.1 `NppStatus nppiFilterGauss_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

Four channel 16-bit signed Gauss filter, ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.2 NppStatus nppiFilterGauss\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 16-bit signed Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.3 NppStatus nppiFilterGauss\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit signed Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.4 NppStatus nppiFilterGauss\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit signed Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.5 NppStatus nppiFilterGauss\_16u\_AC4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit unsigned Gauss filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.6 NppStatus nppiFilterGauss\_16u\_C1R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 16-bit unsigned Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.7 **NppStatus nppiFilterGauss\_16u\_C3R** (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit unsigned Gauss filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.8 **NppStatus nppiFilterGauss\_16u\_C4R** (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned Gauss filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.9 **NppStatus nppiFilterGauss\_32f\_AC4R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point Gauss filter, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.10 NppStatus nppiFilterGauss\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 32-bit floating-point Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.11 NppStatus nppiFilterGauss\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Three channel 32-bit floating-point Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.12 NppStatus nppiFilterGauss\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 32-bit floating-point Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.65.2.13 **NppStatus nppiFilterGauss\_8u\_AC4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned Gauss filter, ignoring alpha channel.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.65.2.14 **NppStatus nppiFilterGauss\_8u\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned Gauss filter.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.65.2.15 **NppStatus nppiFilterGauss\_8u\_C3R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 8-bit unsigned Gauss filter.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.65.2.16 NppStatus nppiFilterGauss\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 8-bit unsigned Gauss filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.17 NppStatus nppiFilterHighPass\_16s\_AC4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit signed high-pass filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.18 NppStatus nppiFilterHighPass\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 16-bit signed high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.19 NppStatus nppiFilterHighPass\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 16-bit signed high-pass filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.20 NppStatus nppiFilterHighPass\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed high-pass filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.21 NppStatus nppiFilterHighPass\_16u\_AC4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit unsigned high-pass filter, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.22 NppStatus nppiFilterHighPass\_16u\_C1R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 16-bit unsigned high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.23 NppStatus nppiFilterHighPass\_16u\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Three channel 16-bit unsigned high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.24 NppStatus nppiFilterHighPass\_16u\_C4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit unsigned high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.25 NppStatus nppiFilterHighPass\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.26 NppStatus nppiFilterHighPass\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 32-bit floating-point high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.27 NppStatus nppiFilterHighPass\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Three channel 32-bit floating-point high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.28 NppStatus nppiFilterHighPass\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 32-bit floating-point high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.29 NppStatus nppiFilterHighPass\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 8-bit unsigned high-pass filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.30 NppStatus nppiFilterHighPass\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 8-bit unsigned high-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.31 **NppStatus nppiFilterHighPass\_8u\_C3R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 8-bit unsigned high-pass filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.32 **NppStatus nppiFilterHighPass\_8u\_C4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned high-pass filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.33 **NppStatus nppiFilterLaplace\_16s\_AC4R** (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 16-bit signed Laplace filter, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.34 NppStatus nppiFilterLaplace\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 16-bit signed Laplace filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.35 NppStatus nppiFilterLaplace\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Three channel 16-bit signed Laplace filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.36 NppStatus nppiFilterLaplace\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit signed Laplace filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.37 **NppStatus nppiFilterLaplace\_32f\_AC4R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point Laplace filter, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.38 **NppStatus nppiFilterLaplace\_32f\_C1R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 32-bit floating-point Laplace filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.65.2.39 **NppStatus nppiFilterLaplace\_32f\_C3R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 32-bit floating-point Laplace filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.65.2.40 NppStatus nppiFilterLaplace\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 32-bit floating-point Laplace filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.41 NppStatus nppiFilterLaplace\_8s16s\_C1R (const Npp8s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 8-bit signed to 16-bit signed Laplace filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.42 NppStatus nppiFilterLaplace\_8u16s\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 8-bit unsigned to 16-bit signed Laplace filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.65.2.43 **NppStatus nppiFilterLaplace\_8u\_AC4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned Laplace filter, ignoring alpha channel.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.65.2.44 **NppStatus nppiFilterLaplace\_8u\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned Laplace filter.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.65.2.45 **NppStatus nppiFilterLaplace\_8u\_C3R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Three channel 8-bit unsigned Laplace filter.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.46** `NppStatus nppiFilterLaplace_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Four channel 8-bit unsigned Laplace filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.47** `NppStatus nppiFilterLowPass_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Four channel 16-bit signed low-pass filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.48** `NppStatus nppiFilterLowPass_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 16-bit signed low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.49 NppStatus nppiFilterLowPass\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Three channel 16-bit signed low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.50 NppStatus nppiFilterLowPass\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit signed low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.51 NppStatus nppiFilterLowPass\_16u\_AC4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit unsigned low-pass filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.52 NppStatus nppiFilterLowPass\_16u\_C1R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 16-bit unsigned low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.53 NppStatus nppiFilterLowPass\_16u\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Three channel 16-bit unsigned low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.54 NppStatus nppiFilterLowPass\_16u\_C4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 16-bit unsigned low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.55 NppStatus nppiFilterLowPass\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Four channel 32-bit floating-point high-pass filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.56 NppStatus nppiFilterLowPass\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 32-bit floating-point low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.57 NppStatus nppiFilterLowPass\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Three channel 32-bit floating-point low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.58 NppStatus nppiFilterLowPass\_32f\_C4R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 32-bit floating-point low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.59 NppStatus nppiFilterLowPass\_8u\_AC4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Four channel 8-bit unsigned low-pass filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.60 NppStatus nppiFilterLowPass\_8u\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.61** `NppStatus nppiFilterLowPass_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Three channel 8-bit unsigned low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.62** `NppStatus nppiFilterLowPass_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Four channel 8-bit unsigned low-pass filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.63** `NppStatus nppiFilterRobertsDown_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed horizontal Roberts filter, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.65.2.64 NppStatus nppiFilterRobertsDown\_16s\_C1R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 16-bit signed horizontal Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.65 NppStatus nppiFilterRobertsDown\_16s\_C3R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Three channel 16-bit signed horizontal Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.66 NppStatus nppiFilterRobertsDown\_16s\_C4R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Four channel 16-bit signed horizontal Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.67 **NppStatus nppiFilterRobertsDown\_32f\_AC4R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point horizontal Roberts filter, ignoring alpha-channel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.68 **NppStatus nppiFilterRobertsDown\_32f\_C1R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point horizontal Roberts filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.69 **NppStatus nppiFilterRobertsDown\_32f\_C3R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating-point horizontal Roberts filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.70 NppStatus nppiFilterRobertsDown\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point horizontal Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.71 NppStatus nppiFilterRobertsDown\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned horizontal Roberts filter, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.72 NppStatus nppiFilterRobertsDown\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned horizontal Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.73 NppStatus nppiFilterRobertsDown\_8u\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned horizontal Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.74 NppStatus nppiFilterRobertsDown\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned horizontal Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.75 NppStatus nppiFilterRobertsUp\_16s\_AC4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed vertical Roberts filter, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.76 NppStatus nppiFilterRobertsUp\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 16-bit signed vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.77 NppStatus nppiFilterRobertsUp\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 16-bit signed vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.78 NppStatus nppiFilterRobertsUp\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.79 NppStatus nppiFilterRobertsUp\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point vertical Roberts filter, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.80 NppStatus nppiFilterRobertsUp\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 32-bit floating-point vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.81 NppStatus nppiFilterRobertsUp\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 32-bit floating-point vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.82 NppStatus nppiFilterRobertsUp\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.83 NppStatus nppiFilterRobertsUp\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned vertical Roberts filter, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.84 NppStatus nppiFilterRobertsUp\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.85 NppStatus nppiFilterRobertsUp\_8u\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.86 NppStatus nppiFilterRobertsUp\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned vertical Roberts filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.87 NppStatus nppiFilterSharpen\_16s\_AC4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed sharpening filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.65.2.88 NppStatus nppiFilterSharpen\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 16-bit signed sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.89 NppStatus nppiFilterSharpen\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 16-bit signed sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.90 NppStatus nppiFilterSharpen\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.91 **NppStatus nppiFilterSharpen\_16u\_AC4R** (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 16-bit unsigned sharpening filter, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.92 **NppStatus nppiFilterSharpen\_16u\_C1R** (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 16-bit unsigned sharpening filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.65.2.93 **NppStatus nppiFilterSharpen\_16u\_C3R** (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 16-bit unsigned sharpening filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.94 NppStatus nppiFilterSharpen\_16u\_C4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit unsigned sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.95 NppStatus nppiFilterSharpen\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point sharpening filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.96 NppStatus nppiFilterSharpen\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 32-bit floating-point sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.97 NppStatus nppiFilterSharpen\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 32-bit floating-point sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.98 NppStatus nppiFilterSharpen\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.99 NppStatus nppiFilterSharpen\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned sharpening filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.100 NppStatus nppiFilterSharpen\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.101 NppStatus nppiFilterSharpen\_8u\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.102 NppStatus nppiFilterSharpen\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned sharpening filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.65.2.103 NppStatus nppiFilterSobelCrossBorder\_32f\_C1R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)

Single channel 32-bit floating-point second cross derivative Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to *pSrc*.

*oSrcOffset* Source image starting point relative to *pSrc*.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.104 NppStatus nppiFilterSobelCrossBorder\_8s16s\_C1R** (const Npp8s \* *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to *pSrc*.

*oSrcOffset* Source image starting point relative to *pSrc*.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.105** `NppStatus nppiFilterSobelCrossBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.106** `NppStatus nppiFilterSobelVertSecondBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point second derivative, vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.107 NppStatus nppiFilterSobelVertSecondBorder\_8s16s\_C1R (const Npp8s \* *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)**

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to *pSrc*.

*oSrcOffset* Source image starting point relative to *pSrc*.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.65.2.108 NppStatus nppiFilterSobelVertSecondBorder\_8u16s\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*, NppiBorderType *eBorderType*)**

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to *pSrc*.

*oSrcOffset* Source image starting point relative to *pSrc*.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



## 7.66 1D Linear Filter

### Modules

- [1D Window Sum](#)
- [Convolution](#)
- [2D Fixed Linear Filters](#)
- [Rank Filters](#)
- [Fixed Filters](#)

*Fixed filters perform linear filtering operations (i.e.*

### FilterColumn

Apply convolution filter with user specified 1D column of weights.

Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring column pixel values in the source image defined by nKernelDim and nAnchorY, divided by nDivisor.

- [NppStatus nppiFilterColumn\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) \*pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

*8-bit unsigned single-channel 1D column convolution.*

- [NppStatus nppiFilterColumn\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) \*pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

*8-bit unsigned three-channel 1D column convolution.*

- [NppStatus nppiFilterColumn\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) \*pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

*8-bit unsigned four-channel 1D column convolution.*

- [NppStatus nppiFilterColumn\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp8u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) \*pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

*8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.*

- [NppStatus nppiFilterColumn\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp16u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) \*pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

*16-bit unsigned single-channel 1D column convolution.*

- [NppStatus nppiFilterColumn\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc, [Npp32s](#) nSrcStep, [Npp16u](#) \*pDst, [Npp32s](#) nDstStep, [NppiSize](#) oROI, const [Npp32s](#) \*pKernel, [Npp32s](#) nMaskSize, [Npp32s](#) nAnchor, [Npp32s](#) nDivisor)

*16-bit unsigned three-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit unsigned four-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.*

- `NppStatus nppiFilterColumn_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit single-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit three-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit four-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit four-channel 1D column convolution ignoring alpha-channel.*

- `NppStatus nppiFilterColumn_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*32-bit float single-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*32-bit float three-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*32-bit float four-channel 1D column convolution.*

- `NppStatus nppiFilterColumn_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*32-bit float four-channel 1D column convolution ignoring alpha-channel.*

- `NppStatus nppiFilterColumn_64f_C1R` (const `Npp64f` \*pSrc, `Npp32s` nSrcStep, `Npp64f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp64f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*64-bit float single-channel 1D column convolution.*

**FilterColumn32f**

FilterColumn using floating-point weights.

- `NppStatus nppiFilterColumn32f_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*8-bit unsigned single-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*8-bit unsigned three-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*8-bit unsigned four-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.*
- `NppStatus nppiFilterColumn32f_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit unsigned single-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit unsigned three-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit unsigned four-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.*
- `NppStatus nppiFilterColumn32f_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit single-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit three-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit four-channel 1D column convolution.*
- `NppStatus nppiFilterColumn32f_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*16-bit four-channel 1D column convolution ignoring alpha-channel.*

*16-bit four-channel 1D column convolution ignoring alpha-channel.*

## FilterRow

Apply convolution filter with user specified 1D row of weights.

Result pixel is equal to the sum of the products between the kernel coefficients (pKernel array) and corresponding neighboring row pixel values in the source image defined by nKernelDim and nAnchorX, divided by nDivisor.

- `NppStatus nppiFilterRow_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*8-bit unsigned single-channel 1D row convolution.*

- `NppStatus nppiFilterRow_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*8-bit unsigned three-channel 1D row convolution.*

- `NppStatus nppiFilterRow_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*8-bit unsigned four-channel 1D row convolution.*

- `NppStatus nppiFilterRow_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.*

- `NppStatus nppiFilterRow_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit unsigned single-channel 1D row convolution.*

- `NppStatus nppiFilterRow_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit unsigned three-channel 1D row convolution.*

- `NppStatus nppiFilterRow_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit unsigned four-channel 1D row convolution.*

- `NppStatus nppiFilterRow_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32s` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor, `Npp32s` nDivisor)

*16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.*

- **NppStatus nppiFilterRow\_16s\_C1R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32s** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor)

*16-bit single-channel 1D row convolution.*

- **NppStatus nppiFilterRow\_16s\_C3R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32s** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor)

*16-bit three-channel 1D row convolution.*

- **NppStatus nppiFilterRow\_16s\_C4R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32s** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor)

*16-bit four-channel 1D row convolution.*

- **NppStatus nppiFilterRow\_16s\_AC4R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32s** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor, **Npp32s** nDivisor)

*16-bit four-channel 1D row convolution ignoring alpha-channel.*

- **NppStatus nppiFilterRow\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

*32-bit float single-channel 1D row convolution.*

- **NppStatus nppiFilterRow\_32f\_C3R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

*32-bit float three-channel 1D row convolution.*

- **NppStatus nppiFilterRow\_32f\_C4R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

*32-bit float four-channel 1D row convolution.*

- **NppStatus nppiFilterRow\_32f\_AC4R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

*32-bit float four-channel 1D row convolution ignoring alpha-channel.*

- **NppStatus nppiFilterRow\_64f\_C1R** (const **Npp64f** \*pSrc, **Npp32s** nSrcStep, **Npp64f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp64f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

*64-bit float single-channel 1D row convolution.*

## FilterRow32f

FilterRow using floating-point weights.

- **NppStatus nppiFilterRow32f\_8u\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

*8-bit unsigned single-channel 1D row convolution.*

- **NppStatus nppiFilterRow32f\_8u\_C3R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

*8-bit unsigned three-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*8-bit unsigned four-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.*

- `NppStatus nppiFilterRow32f_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit unsigned single-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit unsigned three-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit unsigned four-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.*

- `NppStatus nppiFilterRow32f_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit single-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit three-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit four-channel 1D row convolution.*

- `NppStatus nppiFilterRow32f_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, const `Npp32f` \*pKernel, `Npp32s` nMaskSize, `Npp32s` nAnchor)

*16-bit four-channel 1D row convolution ignoring alpha-channel.*

## FilterSobelVertSecond

Filters the image using a second derivative, vertical Sobel filter kernel:

$$\begin{pmatrix} 1 & -2 & 1 \\ 2 & -4 & 2 \\ 1 & -2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -2 & 0 & 1 \\ 4 & 0 & -8 & 0 & 4 \\ 6 & 0 & -12 & 0 & 6 \\ 4 & 0 & -8 & 0 & 4 \\ 1 & 0 & -2 & 0 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelVertSecond_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter.*
- `NppStatus nppiFilterSobelVertSecond_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter.*
- `NppStatus nppiFilterSobelVertSecond_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 32-bit floating-point second derivative, vertical Sobel filter.*

## FilterSobelCross

Filters the image using a second cross derivative Sobel filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -2 & -4 & 0 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & -4 & -2 \\ 1 & 2 & 0 & -2 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelCross_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter.*
- `NppStatus nppiFilterSobelCross_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter.*
- `NppStatus nppiFilterSobelCross_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 32-bit floating-point second cross derivative Sobel filter.*

## FilterSobelHorizBorder

Filters the image using a horizontal Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelHorizBorder_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)  
*Single channel 8-bit unsigned horizontal Sobel filter with border control.*



- `NppStatus nppiFilterSobelHorizBorder_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Three channel 8-bit unsigned horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 16-bit signed horizontal Sobel filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterSobelHorizBorder_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 16-bit signed horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Three channel 16-bit signed horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 16-bit signed horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned horizontal Sobel filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterSobelHorizBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 32-bit floating-point horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Three channel 32-bit floating-point horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 32-bit floating-point horizontal Sobel filter with border control.*



- `NppStatus nppiFilterSobelHorizBorder_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 32-bit floating-point horizontal Sobel filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterSobelHorizBorder_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizBorder_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit signed to 16-bit signed horizontal Sobel filter with border control.*

- `NppStatus nppiFilterSobelHorizMaskBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 32-bit floating-point horizontal Sobel filter with border control.*

## FilterSobelVertBorder

Filters the image using a vertical Sobel filter kernel with border control:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelVertBorder_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Three channel 8-bit unsigned vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 16-bit signed vertical Sobel filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterSobelVertBorder_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 16-bit signed vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Three channel 16-bit signed vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 16-bit signed vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned vertical Sobel filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterSobelVertBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 32-bit floating-point vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Three channel 32-bit floating-point vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 32-bit floating-point vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Four channel 32-bit floating-point vertical Sobel filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterSobelVertBorder_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter with border control.*

- `NppStatus nppiFilterSobelVertBorder_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize, `NppiBorderType` eBorderType)

*Single channel 8-bit signed to 16-bit signed vertical Sobel filter with border control.*

- **NppStatus** **nppiFilterSobelVertMaskBorder\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

*Single channel 32-bit floating-point vertical Sobel filter with border control.*

## FilterSobelHorizSecondBorder

Filters the image using a second derivative, horizontal Sobel filter kernel with border control:

$$\begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- **NppStatus** **nppiFilterSobelHorizSecondBorder\_8u16s\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

*Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter with border control.*

- **NppStatus** **nppiFilterSobelHorizSecondBorder\_8s16s\_C1R** (const **Npp8s** \*pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

*Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter with border control.*

- **NppStatus** **nppiFilterSobelHorizSecondBorder\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **NppiSize** oSrcSize, **NppiPoint** oSrcOffset, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiMaskSize** eMaskSize, **NppiBorderType** eBorderType)

*Single channel 32-bit floating-point second derivative, horizontal Sobel filter with border control.*

## 7.66.1 Function Documentation

**7.66.1.1 NppStatus nppiFilterColumn32f\_16s\_AC4R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, const **Npp32f** \*pKernel, **Npp32s** nMaskSize, **Npp32s** nAnchor)

16-bit four-channel 1D column convolution ignoring alpha-channel.

### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pDst** Destination-Image Pointer.

**nDstStep** Destination-Image Line Step.

**oROI** Region-of-Interest (ROI).

**pKernel** Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

**nMaskSize** Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.2** `NppStatus nppiFilterColumn32f_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit single-channel 1D column convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.3** `NppStatus nppiFilterColumn32f_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit three-channel 1D column convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.4 NppStatus nppiFilterColumn32f\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

16-bit four-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.5 NppStatus nppiFilterColumn32f\_16u\_AC4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.6 NppStatus nppiFilterColumn32f\_16u\_C1R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

16-bit unsigned single-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.7 NppStatus nppiFilterColumn32f\_16u\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

16-bit unsigned three-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.8 NppStatus nppiFilterColumn32f\_16u\_C4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

16-bit unsigned four-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.9 NppStatus nppiFilterColumn32f\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.10** `NppStatus nppiFilterColumn32f_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned single-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.11** `NppStatus nppiFilterColumn32f_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned three-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.66.1.12** `NppStatus nppiFilterColumn32f_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned four-channel 1D column convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.13** `NppStatus nppiFilterColumn_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit four-channel 1D column convolution ignoring alpha-channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.14 NppStatus nppiFilterColumn\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32s \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*, Npp32s *nDivisor*)**

16-bit single-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.15 NppStatus nppiFilterColumn\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32s \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*, Npp32s *nDivisor*)**

16-bit three-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.16** `NppStatus nppiFilterColumn_16s_C4R` (const `Npp16s * pSrc`, `Npp32s nSrcStep`, `Npp16s * pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32s * pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`)

16-bit four-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.17** `NppStatus nppiFilterColumn_16u_AC4R` (const `Npp16u * pSrc`, `Npp32s nSrcStep`, `Npp16u * pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32s * pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`, `Npp32s nDivisor`)

16-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.18** `NppStatus nppiFilterColumn_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned single-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.19** `NppStatus nppiFilterColumn_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned three-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.20** `NppStatus nppiFilterColumn_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned four-channel 1D column convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.21** `NppStatus nppiFilterColumn_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float four-channel 1D column convolution ignoring alpha-channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.66.1.22 **NppStatus nppiFilterColumn\_32f\_C1R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

32-bit float single-channel 1D column convolution.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.66.1.23 **NppStatus nppiFilterColumn\_32f\_C3R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

32-bit float three-channel 1D column convolution.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.24 NppStatus nppiFilterColumn\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

32-bit float four-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.25 NppStatus nppiFilterColumn\_64f\_C1R (const Npp64f \* *pSrc*, Npp32s *nSrcStep*, Npp64f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp64f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

64-bit float single-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.26** `NppStatus nppiFilterColumn_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D column convolution ignoring alpha-channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.27** `NppStatus nppiFilterColumn_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned single-channel 1D column convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.66.1.28** `NppStatus nppiFilterColumn_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned three-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.29** `NppStatus nppiFilterColumn_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D column convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.30** `NppStatus nppiFilterRow32f_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit four-channel 1D row convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.31** `NppStatus nppiFilterRow32f_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit single-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.32 NppStatus nppiFilterRow32f\_16s\_C3R** (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit three-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.33 NppStatus nppiFilterRow32f\_16s\_C4R** (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

16-bit four-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.34 NppStatus nppiFilterRow32f\_16u\_AC4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.35 NppStatus nppiFilterRow32f\_16u\_C1R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, const Npp32f \* *pKernel*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

16-bit unsigned single-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.36** `NppStatus nppiFilterRow32f_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit unsigned three-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.37** `NppStatus nppiFilterRow32f_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

16-bit unsigned four-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.38** `NppStatus nppiFilterRow32f_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.39** `NppStatus nppiFilterRow32f_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned single-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.40** `NppStatus nppiFilterRow32f_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned three-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.41** `NppStatus nppiFilterRow32f_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

8-bit unsigned four-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.42** `NppStatus nppiFilterRow_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit four-channel 1D row convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.43** `NppStatus nppiFilterRow_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit single-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.66.1.44** `NppStatus nppiFilterRow_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit three-channel 1D row convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.45** `NppStatus nppiFilterRow_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit four-channel 1D row convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.46** `NppStatus nppiFilterRow_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.47** `NppStatus nppiFilterRow_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned single-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.48** `NppStatus nppiFilterRow_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned three-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.49** `NppStatus nppiFilterRow_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

16-bit unsigned four-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.50** `NppStatus nppiFilterRow_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float four-channel 1D row convolution ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.51** `NppStatus nppiFilterRow_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

32-bit float single-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.52** `NppStatus nppiFilterRow_32f_C3R` (const `Npp32f * pSrc`, `Npp32s nSrcStep`, `Npp32f * pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32f * pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`)

32-bit float three-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.53** `NppStatus nppiFilterRow_32f_C4R` (const `Npp32f * pSrc`, `Npp32s nSrcStep`, `Npp32f * pDst`, `Npp32s nDstStep`, `NppiSize oROI`, const `Npp32f * pKernel`, `Npp32s nMaskSize`, `Npp32s nAnchor`)

32-bit float four-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.54** `NppStatus nppiFilterRow_64f_C1R (const Npp64f * pSrc, Npp32s nSrcStep, Npp64f * pDst, Npp32s nDstStep, NppiSize oROI, const Npp64f * pKernel, Npp32s nMaskSize, Npp32s nAnchor)`

64-bit float single-channel 1D row convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.55** `NppStatus nppiFilterRow_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D row convolution ignoring alpha-channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.56** `NppStatus nppiFilterRow_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned single-channel 1D row convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.57** `NppStatus nppiFilterRow_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned three-channel 1D row convolution.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.58** `NppStatus nppiFilterRow_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oROI, const Npp32s * pKernel, Npp32s nMaskSize, Npp32s nAnchor, Npp32s nDivisor)`

8-bit unsigned four-channel 1D row convolution.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.59** `NppStatus nppiFilterSobelCross_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point second cross derivative Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.60** `NppStatus nppiFilterSobelCross_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit signed to 16-bit signed second cross derivative Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.66.1.61 NppStatus nppiFilterSobelCross\_8u16s\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)

Single channel 8-bit unsigned to 16-bit signed second cross derivative Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.66.1.62 NppStatus nppiFilterSobelHorizBorder\_16s\_AC4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, NppiSize *oSrcSize*, NppiPoint *oSrcOffset*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiBorderType *eBorderType*)

Four channel 8-bit unsigned horizontal Sobel filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Source image width and height in pixels relative to *pSrc*.  
*oSrcOffset* Source image starting point relative to *pSrc*.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.63** `NppStatus nppiFilterSobelHorizBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.64** `NppStatus nppiFilterSobelHorizBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.65** `NppStatus nppiFilterSobelHorizBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.66** `NppStatus nppiFilterSobelHorizBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point horizontal Sobel filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.67** `NppStatus nppiFilterSobelHorizBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.68** `NppStatus nppiFilterSobelHorizBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point horizontal Sobel filter with border control.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.69** `NppStatus nppiFilterSobelHorizBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point horizontal Sobel filter with border control.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.70** `NppStatus nppiFilterSobelHorizBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.71** `NppStatus nppiFilterSobelHorizBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.72** `NppStatus nppiFilterSobelHorizBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed horizontal Sobel filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.73** `NppStatus nppiFilterSobelHorizBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.74** `NppStatus nppiFilterSobelHorizBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.75** `NppStatus nppiFilterSobelHorizBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.76** `NppStatus nppiFilterSobelHorizMaskBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.77** `NppStatus nppiFilterSobelHorizSecondBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point second derivative, horizontal Sobel filter with border control.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.78** `NppStatus nppiFilterSobelHorizSecondBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter with border control.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).



*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.79** `NppStatus nppiFilterSobelHorizSecondBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.80** `NppStatus nppiFilterSobelVertBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Sobel filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.81** `NppStatus nppiFilterSobelVertBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 16-bit signed vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.82** `NppStatus nppiFilterSobelVertBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 16-bit signed vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.83** `NppStatus nppiFilterSobelVertBorder_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.84** `NppStatus nppiFilterSobelVertBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point vertical Sobel filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.85** `NppStatus nppiFilterSobelVertBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.86** `NppStatus nppiFilterSobelVertBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 32-bit floating-point vertical Sobel filter with border control.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.87** `NppStatus nppiFilterSobelVertBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 32-bit floating-point vertical Sobel filter with border control.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.88** `NppStatus nppiFilterSobelVertBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.89** `NppStatus nppiFilterSobelVertBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.90** `NppStatus nppiFilterSobelVertBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 16-bit signed vertical Sobel filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.91** `NppStatus nppiFilterSobelVertBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.92** `NppStatus nppiFilterSobelVertBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Three channel 8-bit unsigned vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.93** `NppStatus nppiFilterSobelVertBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Four channel 8-bit unsigned vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.66.1.94** `NppStatus nppiFilterSobelVertMaskBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize, NppiBorderType eBorderType)`

Single channel 32-bit floating-point vertical Sobel filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* Enumeration value specifying the mask size.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.95** `NppStatus nppiFilterSobelVertSecond_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point second derivative, vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.66.1.96** `NppStatus nppiFilterSobelVertSecond_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit signed to 16-bit signed second derivative, vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.66.1.97** `NppStatus nppiFilterSobelVertSecond_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit unsigned to 16-bit signed second derivative, vertical Sobel filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.67 1D Window Sum

### 1D Window Sum

1D mask Window Sum for 8 and 16 bit images.

- `NppStatus nppiSumWindowColumn_8u32f_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*One channel 8-bit unsigned 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_8u32f_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*Three channel 8-bit unsigned 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_8u32f_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*Four channel 8-bit unsigned 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_16u32f_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*One channel 16-bit unsigned 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_16u32f_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*Three channel 16-bit unsigned 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_16u32f_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*Four channel 16-bit unsigned 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_16s32f_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*One channel 16-bit signed 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_16s32f_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*Three channel 16-bit signed 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowColumn_16s32f_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*Four channel 16-bit signed 1D (column) sum to 32f.*
- `NppStatus nppiSumWindowRow_8u32f_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*One channel 8-bit unsigned 1D (row) sum to 32f.*
- `NppStatus nppiSumWindowRow_8u32f_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oROI, `Npp32s` nMaskSize, `Npp32s` nAnchor)  
*Three channel 8-bit unsigned 1D (row) sum to 32f.*

- **NppStatus nppiSumWindowRow\_8u32f\_C4R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)  
*Four channel 8-bit unsigned 1D (row) sum to 32f.*
- **NppStatus nppiSumWindowRow\_16u32f\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)  
*One channel 16-bit unsigned 1D (row) sum to 32f.*
- **NppStatus nppiSumWindowRow\_16u32f\_C3R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)  
*Three channel 16-bit unsigned 1D (row) sum to 32f.*
- **NppStatus nppiSumWindowRow\_16u32f\_C4R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)  
*Four channel 16-bit unsigned 1D (row) sum to 32f.*
- **NppStatus nppiSumWindowRow\_16s32f\_C1R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)  
*One channel 16-bit signed 1D (row) sum to 32f.*
- **NppStatus nppiSumWindowRow\_16s32f\_C3R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)  
*Three channel 16-bit signed 1D (row) sum to 32f.*
- **NppStatus nppiSumWindowRow\_16s32f\_C4R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oROI, **Npp32s** nMaskSize, **Npp32s** nAnchor)  
*Four channel 16-bit signed 1D (row) sum to 32f.*

## 7.67.1 Function Documentation

### 7.67.1.1 NppStatus nppiSumWindowColumn\_16s32f\_C1R (const Npp16s \*pSrc, Npp32s nSrcStep, Npp32f \*pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)

One channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

#### Parameters:

- pSrc** Source-Image Pointer.
- nSrcStep** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oROI** Region-of-Interest (ROI).
- nMaskSize** Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.2 NppStatus nppiSumWindowColumn\_16s32f\_C3R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp32f \* pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)**

Three channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.3 NppStatus nppiSumWindowColumn\_16s32f\_C4R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp32f \* pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)**

Four channel 16-bit signed 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.4 NppStatus nppiSumWindowColumn\_16u32f\_C1R (const Npp16u \* pSrc, Npp32s nSrcStep, Npp32f \* pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)**

One channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.5 NppStatus nppiSumWindowColumn\_16u32f\_C3R (const Npp16u \* pSrc, Npp32s nSrcStep, Npp32f \* pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)**

Three channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.6** `NppStatus nppiSumWindowColumn_16u32f_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)`

Four channel 16-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 16 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.7** `NppStatus nppiSumWindowColumn_8u32f_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oROI, Npp32s nMaskSize, Npp32s nAnchor)`

One channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 1-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by nMaskSize and nAnchor.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.8 NppStatus nppiSumWindowColumn\_8u32f\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

Three channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 3-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by *nMaskSize* and *nAnchor*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.9 NppStatus nppiSumWindowColumn\_8u32f\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

Four channel 8-bit unsigned 1D (column) sum to 32f.

Apply Column Window Summation filter over a 1D mask region around each source pixel for 4-channel 8 bit/pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring column pixel values in a mask region of the source image defined by *nMaskSize* and *nAnchor*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* Y offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.10 NppStatus nppiSumWindowRow\_16s32f\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

One channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.11 NppStatus nppiSumWindowRow\_16s32f\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

Three channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.67.1.12 NppStatus nppiSumWindowRow\_16s32f\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

Four channel 16-bit signed 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.13 NppStatus nppiSumWindowRow\_16u32f\_C1R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

One channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.14 NppStatus nppiSumWindowRow\_16u32f\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

Three channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.15 NppStatus nppiSumWindowRow\_16u32f\_C4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)**

Four channel 16-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 16-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by *iKernelDim* and *iAnchorX*.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.16 NppStatus nppiSumWindowRow\_8u32f\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

One channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 1-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.17 NppStatus nppiSumWindowRow\_8u32f\_C3R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Three channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 3-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.67.1.18 NppStatus nppiSumWindowRow\_8u32f\_C4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oROI*, Npp32s *nMaskSize*, Npp32s *nAnchor*)

Four channel 8-bit unsigned 1D (row) sum to 32f.

Apply Row Window Summation filter over a 1D mask region around each source pixel for 4-channel 8-bit pixel input images with 32-bit floating point output. Result 32-bit floating point pixel is equal to the sum of the corresponding and neighboring row pixel values in a mask region of the source image defined by iKernelDim and iAnchorX.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nMaskSize* Length of the linear kernel array.

*nAnchor* X offset of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.68 Convolution

### Filter

General purpose 2D convolution filter.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor.

- `NppStatus nppiFilter_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Single channel 8-bit unsigned convolution filter.*

- `NppStatus nppiFilter_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Three channel 8-bit unsigned convolution filter.*

- `NppStatus nppiFilter_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Four channel channel 8-bit unsigned convolution filter.*

- `NppStatus nppiFilter_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Four channel 8-bit unsigned convolution filter, ignoring alpha channel.*

- `NppStatus nppiFilter_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Single channel 16-bit unsigned convolution filter.*

- `NppStatus nppiFilter_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Three channel 16-bit unsigned convolution filter.*

- `NppStatus nppiFilter_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Four channel channel 16-bit unsigned convolution filter.*

- `NppStatus nppiFilter_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Four channel 16-bit unsigned convolution filter, ignoring alpha channel.*

- `NppStatus nppiFilter_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Single channel 16-bit convolution filter.*

- `NppStatus nppiFilter_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Three channel 16-bit convolution filter.*

- `NppStatus nppiFilter_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Four channel 16-bit convolution filter.*

- `NppStatus nppiFilter_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor)

*Four channel 16-bit convolution filter, ignoring alpha channel.*

- `NppStatus nppiFilter_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Single channel 32-bit float convolution filter.*

- `NppStatus nppiFilter_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Three channel 32-bit float convolution filter.*

- `NppStatus nppiFilter_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 32-bit float convolution filter.*

- `NppStatus nppiFilter_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 32-bit float convolution filter, ignoring alpha channel.*

- `NppStatus nppiFilter_64f_C1R` (const `Npp64f` \*pSrc, `Npp32s` nSrcStep, `Npp64f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp64f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Single channel 64-bit float convolution filter.*

## Filter32f

General purpose 2D convolution filter using floating-point weights.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor.

- `NppStatus nppiFilter32f_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Single channel 8-bit unsigned convolution filter.*

- `NppStatus nppiFilter32f_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Three channel 8-bit unsigned convolution filter.*

- `NppStatus nppiFilter32f_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 8-bit unsigned convolution filter.*

- `NppStatus nppiFilter32f_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 8-bit unsigned convolution filter, ignoring alpha channel.*

- `NppStatus nppiFilter32f_8s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Single channel 8-bit signed convolution filter.*

- `NppStatus nppiFilter32f_8s_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Three channel 8-bit signed convolution filter.*

- `NppStatus nppiFilter32f_8s_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 8-bit signed convolution filter.*

- `NppStatus nppiFilter32f_8s_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 8-bit signed convolution filter, ignoring alpha channel.*

- `NppStatus nppiFilter32f_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Single channel 16-bit unsigned convolution filter.*

- `NppStatus nppiFilter32f_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Three channel 16-bit unsigned convolution filter.*

- `NppStatus nppiFilter32f_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 16-bit unsigned convolution filter.*

- `NppStatus nppiFilter32f_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 16-bit unsigned convolution filter, ignoring alpha channel.*

- `NppStatus nppiFilter32f_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Single channel 16-bit convolution filter.*

- `NppStatus nppiFilter32f_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Three channel 16-bit convolution filter.*

- `NppStatus nppiFilter32f_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Four channel 16-bit convolution filter.*
- `NppStatus nppiFilter32f_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Four channel 16-bit convolution filter, ignoring alpha channel.*
- `NppStatus nppiFilter32f_32s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Single channel 32-bit convolution filter.*
- `NppStatus nppiFilter32f_32s_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Three channel 32-bit convolution filter.*
- `NppStatus nppiFilter32f_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Four channel 32-bit convolution filter.*
- `NppStatus nppiFilter32f_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Four channel 32-bit convolution filter, ignoring alpha channel.*
- `NppStatus nppiFilter32f_8u16s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Single channel 8-bit unsigned to 16-bit signed convolution filter.*
- `NppStatus nppiFilter32f_8u16s_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Three channel 8-bit unsigned to 16-bit signed convolution filter.*
- `NppStatus nppiFilter32f_8u16s_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Four channel 8-bit unsigned to 16-bit signed convolution filter.*
- `NppStatus nppiFilter32f_8u16s_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Four channel 8-bit unsigned to 16-bit signed convolution filter, ignoring alpha channel.*
- `NppStatus nppiFilter32f_8s16s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Single channel 8-bit to 16-bit signed convolution filter.*
- `NppStatus nppiFilter32f_8s16s_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Three channel 8-bit to 16-bit signed convolution filter.*
- `NppStatus nppiFilter32f_8s16s_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)  
*Four channel 8-bit to 16-bit signed convolution filter.*



- `NppStatus nppiFilter32f_8s16s_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor)

*Four channel 8-bit to 16-bit signed convolution filter, ignoring alpha channel.*

## FilterBorder

General purpose 2D convolution filter with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported.

Note: The new Border versions of all Filter functions are in beta test in the 6.0 release of NPP.

- `NppStatus nppiFilterBorder_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Three channel 8-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Four channel channel 8-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Single channel 16-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Three channel 16-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Four channel channel 16-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Single channel 16-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Three channel 16-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Four channel channel 16-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32s` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `Npp32s` nDivisor, `NppiBorderType` eBorderType)

*Four channel 16-bit convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 32-bit float convolution filter with border control.*

- `NppStatus nppiFilterBorder_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 32-bit float convolution filter with border control.*

- `NppStatus nppiFilterBorder_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 32-bit float convolution filter with border control.*

- `NppStatus nppiFilterBorder_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 32-bit float convolution filter with border control, ignoring alpha channel.*

## FilterBorder32f

General purpose 2D convolution filter using floating-point weights with border control.

Pixels under the mask are multiplied by the respective weights in the mask and the results are summed. Before writing the result pixel the sum is scaled back via division by nDivisor. If any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported.

- `NppStatus nppiFilterBorder32f_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 8-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder32f_8s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 8-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8s_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 8-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8s_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 8-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8s_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp8s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 8-bit signed convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder32f_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 16-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 16-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 16-bit unsigned convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder32f_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 16-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 16-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 16-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 16-bit convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder32f_32s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 32-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_32s_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 32-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 32-bit convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 32-bit convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder32f_8u16s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned to 16-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8u16s_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 8-bit unsigned to 16-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8u16s_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned to 16-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8u16s_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 8-bit unsigned to 16-bit signed convolution filter with border control, ignoring alpha channel.*

- `NppStatus nppiFilterBorder32f_8s16s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Single channel 8-bit to 16-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8s16s_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Three channel 8-bit to 16-bit signed convolution filter with border control.*

- `NppStatus nppiFilterBorder32f_8s16s_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` \*pKernel, `NppiSize` oKernelSize, `NppiPoint` oAnchor, `NppiBorderType` eBorderType)

*Four channel 8-bit to 16-bit signed convolution filter with border control.*

- **NppStatus nppiFilterBorder32f\_8s16s\_AC4R** (const [Npp8s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcSize, [NppiPoint](#) oSrcOffset, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp32f](#) \*pKernel, [NppiSize](#) oKernelSize, [NppiPoint](#) oAnchor, [NppiBorderType](#) eBorderType)

*Four channel 8-bit to 16-bit signed convolution filter with border control, ignoring alpha channel.*

## 7.68.1 Function Documentation

- 7.68.1.1 NppStatus nppiFilter32f\_16s\_AC4R** (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp32f](#) \*pKernel, [NppiSize](#) oKernelSize, [NppiPoint](#) oAnchor)

Four channel 16-bit convolution filter, ignoring alpha channel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.68.1.2 NppStatus nppiFilter32f\_16s\_C1R** (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp32f](#) \*pKernel, [NppiSize](#) oKernelSize, [NppiPoint](#) oAnchor)

Single channel 16-bit convolution filter.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.3 NppStatus nppiFilter32f\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)**

Three channel 16-bit convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.4 NppStatus nppiFilter32f\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)**

Four channel 16-bit convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.68.1.5 NppStatus nppiFilter32f\_16u\_AC4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)**

Four channel 16-bit unsigned convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.6 NppStatus nppiFilter32f\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)**

Single channel 16-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.68.1.7 NppStatus nppiFilter32f\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Three channel 16-bit unsigned convolution filter.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.68.1.8 NppStatus nppiFilter32f\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned convolution filter.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.68.1.9 NppStatus nppiFilter32f\_32s\_AC4R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 32-bit convolution filter, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.68.1.10 NppStatus nppiFilter32f\_32s\_C1R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Single channel 32-bit convolution filter.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.11** `NppStatus nppiFilter32f_32s_C3R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 32-bit convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.12** `NppStatus nppiFilter32f_32s_C4R (const Npp32s * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 32-bit convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.13** `NppStatus nppiFilter32f_8s16s_AC4R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit to 16-bit signed convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.14** `NppStatus nppiFilter32f_8s16s_C1R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 8-bit to 16-bit signed convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.68.1.15 `NppStatus nppiFilter32f_8s16s_C3R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 8-bit to 16-bit signed convolution filter.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.68.1.16 `NppStatus nppiFilter32f_8s16s_C4R (const Npp8s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit to 16-bit signed convolution filter.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.17** `NppStatus nppiFilter32f_8s_AC4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit signed convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.18** `NppStatus nppiFilter32f_8s_C1R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 8-bit signed convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.19** `NppStatus nppiFilter32f_8s_C3R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 8-bit signed convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.20** `NppStatus nppiFilter32f_8s_C4R (const Npp8s * pSrc, int nSrcStep, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit signed convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.21** `NppStatus nppiFilter32f_8u16s_AC4R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned to 16-bit signed convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.22** `NppStatus nppiFilter32f_8u16s_C1R (const Npp8u * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 8-bit unsigned to 16-bit signed convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.68.1.23 NppStatus nppiFilter32f\_8u16s\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Three channel 8-bit unsigned to 16-bit signed convolution filter.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.68.1.24 NppStatus nppiFilter32f\_8u16s\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)

Four channel 8-bit unsigned to 16-bit signed convolution filter.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.25** `NppStatus nppiFilter32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned convolution filter, ignorint alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.26** `NppStatus nppiFilter32f_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 8-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.27** `NppStatus nppiFilter32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.28** `NppStatus nppiFilter32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.29** `NppStatus nppiFilter_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel 16-bit convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.30** `NppStatus nppiFilter_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Single channel 16-bit convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.31** `NppStatus nppiFilter_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Three channel 16-bit convolution filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.32** `NppStatus nppiFilter_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel channel 16-bit convolution filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.33** `NppStatus nppiFilter_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel 16-bit unsigned convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.34** `NppStatus nppiFilter_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Single channel 16-bit unsigned convolution filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.35** `NppStatus nppiFilter_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Three channel 16-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.36** `NppStatus nppiFilter_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel channel 16-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.37** `NppStatus nppiFilter_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Four channel 32-bit float convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.38** `NppStatus nppiFilter_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 32-bit float convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.68.1.39 NppStatus nppiFilter\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)**

Three channel 32-bit float convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.40 NppStatus nppiFilter\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, const Npp32f \* *pKernel*, NppiSize *oKernelSize*, NppiPoint *oAnchor*)**

Four channel 32-bit float convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.68.1.41** `NppStatus nppiFilter_64f_C1R (const Npp64f * pSrc, Npp32s nSrcStep, Npp64f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp64f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor)`

Single channel 64-bit float convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.42** `NppStatus nppiFilter_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel 8-bit unsigned convolution filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.43** `NppStatus nppiFilter_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Single channel 8-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.44** `NppStatus nppiFilter_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Three channel 8-bit unsigned convolution filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.45** `NppStatus nppiFilter_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor)`

Four channel channel 8-bit unsigned convolution filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.46** `NppStatus nppiFilterBorder32f_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.47** `NppStatus nppiFilterBorder32f_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 16-bit convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.48** `NppStatus nppiFilterBorder32f_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.49** `NppStatus nppiFilterBorder32f_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.50** `NppStatus nppiFilterBorder32f_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.51** `NppStatus nppiFilterBorder32f_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.52** `NppStatus nppiFilterBorder32f_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.53** `NppStatus nppiFilterBorder32f_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.54** `NppStatus nppiFilterBorder32f_32s_AC4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.68.1.55** `NppStatus nppiFilterBorder32f_32s_C1R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 32-bit convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.56** `NppStatus nppiFilterBorder32f_32s_C3R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 32-bit convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.57** `NppStatus nppiFilterBorder32f_32s_C4R (const Npp32s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.58** `NppStatus nppiFilterBorder32f_8s16s_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit to 16-bit signed convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.59** `NppStatus nppiFilterBorder32f_8s16s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit to 16-bit signed convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.60** `NppStatus nppiFilterBorder32f_8s16s_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit to 16-bit signed convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.61** `NppStatus nppiFilterBorder32f_8s16s_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit to 16-bit signed convolution filter with border control.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* Source image starting point relative to pSrc.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.62** `NppStatus nppiFilterBorder32f_8s_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit signed convolution filter with border control, ignoring alpha channel.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* Source image starting point relative to pSrc.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.63** `NppStatus nppiFilterBorder32f_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit signed convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.64** `NppStatus nppiFilterBorder32f_8s_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit signed convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.65** `NppStatus nppiFilterBorder32f_8s_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit signed convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.66** `NppStatus nppiFilterBorder32f_8u16s_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.67** `NppStatus nppiFilterBorder32f_8u16s_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.68** `NppStatus nppiFilterBorder32f_8u16s_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned to 16-bit signed convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.69** `NppStatus nppiFilterBorder32f_8u16s_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned to 16-bit signed convolution filter with border control.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* Source image starting point relative to pSrc.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.70** `NppStatus nppiFilterBorder32f_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution filter with border control, ignore alpha channel.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Source image width and height in pixels relative to pSrc.
- oSrcOffset* Source image starting point relative to pSrc.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oSizeROI* [Region-of-Interest \(ROI\)](#).
- pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.
- oKernelSize* Width and Height of the rectangular kernel.
- oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.
- eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.68.1.71** `NppStatus nppiFilterBorder32f_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.72** `NppStatus nppiFilterBorder32f_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.73** `NppStatus nppiFilterBorder32f_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.74** `NppStatus nppiFilterBorder_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 16-bit convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.75** `NppStatus nppiFilterBorder_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 16-bit convolution filter with border control.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.76** `NppStatus nppiFilterBorder_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 16-bit convolution filter with border control.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.77 NppStatus nppiFilterBorder\_16s\_C4R (const Npp16s \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s \* pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)**

Four channel channel 16-bit convolution filter with border control.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.78** `NppStatus nppiFilterBorder_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 16-bit unsigned convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.79** `NppStatus nppiFilterBorder_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 16-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided.  
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.80** `NppStatus nppiFilterBorder_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 16-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided.  
If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.81** `NppStatus nppiFilterBorder_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 16-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.82** `NppStatus nppiFilterBorder_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float convolution filter with border control, ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.83** `NppStatus nppiFilterBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Single channel 32-bit float convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.84** `NppStatus nppiFilterBorder_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Three channel 32-bit float convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.  
*oKernelSize* Width and Height of the rectangular kernel.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.68.1.85** `NppStatus nppiFilterBorder_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32f * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, NppiBorderType eBorderType)`

Four channel 32-bit float convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.86** `NppStatus nppiFilterBorder_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel 8-bit unsigned convolution filter with border control, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.87** `NppStatus nppiFilterBorder_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Single channel 8-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.88** `NppStatus nppiFilterBorder_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Three channel 8-bit unsigned convolution filter with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.68.1.89** `NppStatus nppiFilterBorder_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp32s * pKernel, NppiSize oKernelSize, NppiPoint oAnchor, Npp32s nDivisor, NppiBorderType eBorderType)`

Four channel channel 8-bit unsigned convolution filter with border control.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pKernel* Pointer to the start address of the kernel coefficient array. Coefficients are expected to be stored in reverse order.

*oKernelSize* Width and Height of the rectangular kernel.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*nDivisor* The factor by which the convolved summation from the Filter operation should be divided. If equal to the sum of coefficients, this will keep the maximum result value within full scale.

*eBorderType* The border type operation to be applied at source image border boundaries.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.69 2D Fixed Linear Filters

### FilterBox

Computes the average pixel values of the pixels under a rectangular mask.

- `NppStatus nppiFilterBox_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Single channel 8-bit unsigned box filter.*
- `NppStatus nppiFilterBox_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Three channel 8-bit unsigned box filter.*
- `NppStatus nppiFilterBox_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 8-bit unsigned box filter.*
- `NppStatus nppiFilterBox_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 8-bit unsigned box filter, ignoring alpha channel.*
- `NppStatus nppiFilterBox_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Single channel 16-bit unsigned box filter.*
- `NppStatus nppiFilterBox_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Three channel 16-bit unsigned box filter.*
- `NppStatus nppiFilterBox_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 16-bit unsigned box filter.*
- `NppStatus nppiFilterBox_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 16-bit unsigned box filter, ignoring alpha channel.*
- `NppStatus nppiFilterBox_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Single channel 16-bit box filter.*
- `NppStatus nppiFilterBox_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Three channel 16-bit box filter.*
- `NppStatus nppiFilterBox_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 16-bit box filter.*

- `NppStatus nppiFilterBox_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 16-bit box filter, ignoring alpha channel.*

- `NppStatus nppiFilterBox_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Single channel 32-bit floating-point box filter.*

- `NppStatus nppiFilterBox_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Three channel 32-bit floating-point box filter.*

- `NppStatus nppiFilterBox_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 32-bit floating-point box filter.*

- `NppStatus nppiFilterBox_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 32-bit floating-point box filter, ignoring alpha channel.*

- `NppStatus nppiFilterBox_64f_C1R` (const `Npp64f` \*pSrc, `Npp32s` nSrcStep, `Npp64f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Single channel 64-bit floating-point box filter.*

## 7.69.1 Function Documentation

### 7.69.1.1 `NppStatus nppiFilterBox_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

Four channel 16-bit box filter, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.69.1.2 NppStatus nppiFilterBox\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Single channel 16-bit box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.3 NppStatus nppiFilterBox\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Three channel 16-bit box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.4 NppStatus nppiFilterBox\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 16-bit box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.69.1.5** `NppStatus nppiFilterBox_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 16-bit unsigned box filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.69.1.6** `NppStatus nppiFilterBox_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 16-bit unsigned box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.69.1.7 NppStatus nppiFilterBox\_16u\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Three channel 16-bit unsigned box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.8 NppStatus nppiFilterBox\_16u\_C4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 16-bit unsigned box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.9 NppStatus nppiFilterBox\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 32-bit floating-point box filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.10** `NppStatus nppiFilterBox_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 32-bit floating-point box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.11** `NppStatus nppiFilterBox_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 32-bit floating-point box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.12 NppStatus nppiFilterBox\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 32-bit floating-point box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.13 NppStatus nppiFilterBox\_64f\_C1R (const Npp64f \* *pSrc*, Npp32s *nSrcStep*, Npp64f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Single channel 64-bit floating-point box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.69.1.14 NppStatus nppiFilterBox\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 8-bit unsigned box filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.69.1.15** `NppStatus nppiFilterBox_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 8-bit unsigned box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.69.1.16** `NppStatus nppiFilterBox_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.69.1.17** `NppStatus nppiFilterBox_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned box filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Avg operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.70 Rank Filters

### ImageMax Filter

Result pixel value is the maximum of pixel values under the rectangular mask region.

- **NppStatus nppiFilterMax\_8u\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single channel 8-bit unsigned maximum filter.*
- **NppStatus nppiFilterMax\_8u\_C3R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three channel 8-bit unsigned maximum filter.*
- **NppStatus nppiFilterMax\_8u\_C4R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four channel 8-bit unsigned maximum filter.*
- **NppStatus nppiFilterMax\_8u\_AC4R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four channel 8-bit unsigned maximum filter, ignoring alpha channel.*
- **NppStatus nppiFilterMax\_16u\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single channel 16-bit unsigned maximum filter.*
- **NppStatus nppiFilterMax\_16u\_C3R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three channel 16-bit unsigned maximum filter.*
- **NppStatus nppiFilterMax\_16u\_C4R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four channel 16-bit unsigned maximum filter.*
- **NppStatus nppiFilterMax\_16u\_AC4R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four channel 16-bit unsigned maximum filter, ignoring alpha channel.*
- **NppStatus nppiFilterMax\_16s\_C1R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single channel 16-bit signed maximum filter.*
- **NppStatus nppiFilterMax\_16s\_C3R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three channel 16-bit signed maximum filter.*
- **NppStatus nppiFilterMax\_16s\_C4R** (const **Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four channel 16-bit signed maximum filter.*

- `NppStatus nppiFilterMax_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 16-bit signed maximum filter, ignoring alpha channel.*
- `NppStatus nppiFilterMax_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Single channel 32-bit floating-point maximum filter.*
- `NppStatus nppiFilterMax_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Three channel 32-bit floating-point maximum filter.*
- `NppStatus nppiFilterMax_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 32-bit floating-point maximum filter.*
- `NppStatus nppiFilterMax_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 32-bit floating-point maximum filter, ignoring alpha channel.*

## ImageMin Filter

Result pixel value is the minimum of pixel values under the rectangular mask region.

- `NppStatus nppiFilterMin_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Single channel 8-bit unsigned minimum filter.*
- `NppStatus nppiFilterMin_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Three channel 8-bit unsigned minimum filter.*
- `NppStatus nppiFilterMin_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 8-bit unsigned minimum filter.*
- `NppStatus nppiFilterMin_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Four channel 8-bit unsigned minimum filter, ignoring alpha channel.*
- `NppStatus nppiFilterMin_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Single channel 16-bit unsigned minimum filter.*
- `NppStatus nppiFilterMin_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)  
*Three channel 16-bit unsigned minimum filter.*
- `NppStatus nppiFilterMin_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 16-bit unsigned minimum filter.*

- `NppStatus nppiFilterMin_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 16-bit unsigned minimum filter, ignoring alpha channel.*

- `NppStatus nppiFilterMin_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Single channel 16-bit signed minimum filter.*

- `NppStatus nppiFilterMin_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Three channel 16-bit signed minimum filter.*

- `NppStatus nppiFilterMin_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 16-bit signed minimum filter.*

- `NppStatus nppiFilterMin_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 16-bit signed minimum filter, ignoring alpha channel.*

- `NppStatus nppiFilterMin_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Single channel 32-bit floating-point minimum filter.*

- `NppStatus nppiFilterMin_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Three channel 32-bit floating-point minimum filter.*

- `NppStatus nppiFilterMin_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 32-bit floating-point minimum filter.*

- `NppStatus nppiFilterMin_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor)

*Four channel 32-bit floating-point minimum filter, ignoring alpha channel.*

## ImageMedian Filter

Result pixel value is the median of pixel values under the rectangular mask region.

- `NppStatus nppiFilterMedian_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Single channel 8-bit unsigned median filter.*

- `NppStatus nppiFilterMedian_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Three channel 8-bit unsigned median filter.*

- `NppStatus nppiFilterMedian_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 8-bit unsigned median filter.*

- `NppStatus nppiFilterMedian_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 8-bit unsigned median filter, ignoring alpha channel.*

- `NppStatus nppiFilterMedian_16u_C1R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Single channel 16-bit unsigned median filter.*

- `NppStatus nppiFilterMedian_16u_C3R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Three channel 16-bit unsigned median filter.*

- `NppStatus nppiFilterMedian_16u_C4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 16-bit unsigned median filter.*

- `NppStatus nppiFilterMedian_16u_AC4R` (const `Npp16u` \*pSrc, `Npp32s` nSrcStep, `Npp16u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 16-bit unsigned median filter, ignoring alpha channel.*

- `NppStatus nppiFilterMedian_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Single channel 16-bit signed median filter.*

- `NppStatus nppiFilterMedian_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Three channel 16-bit signed median filter.*

- `NppStatus nppiFilterMedian_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 16-bit signed median filter.*

- `NppStatus nppiFilterMedian_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 16-bit signed median filter, ignoring alpha channel.*



- `NppStatus nppiFilterMedian_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Single channel 32-bit floating-point median filter.*

- `NppStatus nppiFilterMedian_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Three channel 32-bit floating-point median filter.*

- `NppStatus nppiFilterMedian_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 32-bit floating-point median filter.*

- `NppStatus nppiFilterMedian_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiSize` oMaskSize, `NppiPoint` oAnchor, `Npp8u` \*pBuffer)

*Four channel 32-bit floating-point median filter, ignoring alpha channel.*

- `NppStatus nppiFilterMedianGetBufferSize_8u_C1R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Single channel 8-bit unsigned median filter scratch memory size.*

- `NppStatus nppiFilterMedianGetBufferSize_8u_C3R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Three channel 8-bit unsigned median filter scratch memory size.*

- `NppStatus nppiFilterMedianGetBufferSize_8u_C4R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Four channel 8-bit unsigned median filter scratch memory size.*

- `NppStatus nppiFilterMedianGetBufferSize_8u_AC4R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Four channel 8-bit unsigned median filter, ignoring alpha channel.*

- `NppStatus nppiFilterMedianGetBufferSize_16u_C1R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Single channel 16-bit unsigned median filter scratch memory size.*

- `NppStatus nppiFilterMedianGetBufferSize_16u_C3R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Three channel 16-bit unsigned median filter scratch memory size.*

- `NppStatus nppiFilterMedianGetBufferSize_16u_C4R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Four channel 16-bit unsigned median filter scratch memory size.*

- `NppStatus nppiFilterMedianGetBufferSize_16u_AC4R` (`NppiSize` oSizeROI, `NppiSize` oMaskSize, `Npp32u` \*nBufferSize)

*Four channel 16-bit unsigned median filter, ignoring alpha channel.*

- **NppStatus nppiFilterMedianGetBufferSize\_16s\_C1R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Single channel 16-bit signed median filter scratch memory size.*
- **NppStatus nppiFilterMedianGetBufferSize\_16s\_C3R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Three channel 16-bit signed median filter scratch memory size.*
- **NppStatus nppiFilterMedianGetBufferSize\_16s\_C4R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Four channel 16-bit signed median filter scratch memory size.*
- **NppStatus nppiFilterMedianGetBufferSize\_16s\_AC4R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Four channel 16-bit signed median filter, ignoring alpha channel.*
- **NppStatus nppiFilterMedianGetBufferSize\_32f\_C1R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Single channel 32-bit floating-point median filter scratch memory size.*
- **NppStatus nppiFilterMedianGetBufferSize\_32f\_C3R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Three channel 32-bit floating-point median filter scratch memory size.*
- **NppStatus nppiFilterMedianGetBufferSize\_32f\_C4R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Four channel 32-bit floating-point median filter scratch memory size.*
- **NppStatus nppiFilterMedianGetBufferSize\_32f\_AC4R** (**NppiSize** oSizeROI, **NppiSize** oMaskSize, **Npp32u** \*nBufferSize)  
*Four channel 32-bit floating-point median filter, ignoring alpha channel.*

## 7.70.1 Function Documentation

### 7.70.1.1 **NppStatus nppiFilterMax\_16s\_AC4R** (**const Npp16s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Four channel 16-bit signed maximum filter, ignoring alpha channel.

#### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pDst** Destination-Image Pointer.

**nDstStep** Destination-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

**oMaskSize** Width and Height of the neighborhood region for the local Max operation.

**oAnchor** X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.70.1.2 **NppStatus nppiFilterMax\_16s\_C1R** (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 16-bit signed maximum filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.70.1.3 **NppStatus nppiFilterMax\_16s\_C3R** (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Three channel 16-bit signed maximum filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.70.1.4 **NppStatus nppiFilterMax\_16s\_C4R** (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit signed maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Max operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.70.1.5 **NppStatus nppiFilterMax\_16u\_AC4R** (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit unsigned maximum filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Max operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.70.1.6 **NppStatus nppiFilterMax\_16u\_C1R** (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 16-bit unsigned maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Max operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.7 NppStatus nppiFilterMax\_16u\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Three channel 16-bit unsigned maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.8 NppStatus nppiFilterMax\_16u\_C4R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 16-bit unsigned maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.9 NppStatus nppiFilterMax\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 32-bit floating-point maximum filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.10** `NppStatus nppiFilterMax_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 32-bit floating-point maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.11** `NppStatus nppiFilterMax_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 32-bit floating-point maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.70.1.12 **NppStatus nppiFilterMax\_32f\_C4R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 32-bit floating-point maximum filter.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.70.1.13 **NppStatus nppiFilterMax\_8u\_AC4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 8-bit unsigned maximum filter, ignoring alpha channel.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.70.1.14 **NppStatus nppiFilterMax\_8u\_C1R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Single channel 8-bit unsigned maximum filter.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.15** `NppStatus nppiFilterMax_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.16** `NppStatus nppiFilterMax_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned maximum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.70.1.17** `NppStatus nppiFilterMedian_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit signed median filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.18** `NppStatus nppiFilterMedian_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 16-bit signed median filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.19** `NppStatus nppiFilterMedian_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 16-bit signed median filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.20** `NppStatus nppiFilterMedian_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit signed median filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.21** `NppStatus nppiFilterMedian_16u_AC4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit unsigned median filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.22** `NppStatus nppiFilterMedian_16u_C1R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 16-bit unsigned median filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.23** `NppStatus nppiFilterMedian_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 16-bit unsigned median filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.24** `NppStatus nppiFilterMedian_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 16-bit unsigned median filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.25** `NppStatus nppiFilterMedian_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 32-bit floating-point median filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.26** `NppStatus nppiFilterMedian_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 32-bit floating-point median filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Median operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.27** `NppStatus nppiFilterMedian_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 32-bit floating-point median filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*oMaskSize* Width and Height of the neighborhood region for the local Median operation.  
*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.  
*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.28** `NppStatus nppiFilterMedian_32f_C4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 32-bit floating-point median filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.29** `NppStatus nppiFilterMedian_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 8-bit unsigned median filter, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.30** `NppStatus nppiFilterMedian_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Single channel 8-bit unsigned median filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.31** `NppStatus nppiFilterMedian_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Three channel 8-bit unsigned median filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.32** `NppStatus nppiFilterMedian_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor, Npp8u * pBuffer)`

Four channel 8-bit unsigned median filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

*pBuffer* Pointer to the user-allocated scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.33** `NppStatus nppiFilterMedianGetBufferSize_16s_AC4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u * nBufferSize)`

Four channel 16-bit signed median filter, ignoring alpha channel.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.34 NppStatus nppiFilterMedianGetBufferSize\_16s\_C1R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u \* nBufferSize)**

Single channel 16-bit signed median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.35 NppStatus nppiFilterMedianGetBufferSize\_16s\_C3R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u \* nBufferSize)**

Three channel 16-bit signed median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.36 NppStatus nppiFilterMedianGetBufferSize\_16s\_C4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u \* nBufferSize)**

Four channel 16-bit signed median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)



**7.70.1.37 NppStatus nppiFilterMedianGetBufferSize\_16u\_AC4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u \* nBufferSize)**

Four channel 16-bit unsigned median filter, ignoring alpha channel.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.38 NppStatus nppiFilterMedianGetBufferSize\_16u\_C1R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u \* nBufferSize)**

Single channel 16-bit unsigned median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.39 NppStatus nppiFilterMedianGetBufferSize\_16u\_C3R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u \* nBufferSize)**

Three channel 16-bit unsigned median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.40 NppStatus nppiFilterMedianGetBufferSize\_16u\_C4R (NppiSize oSizeROI, NppiSize oMaskSize, Npp32u \* nBufferSize)**

Four channel 16-bit unsigned median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

#### 7.70.1.41 **NppStatus nppiFilterMedianGetBufferSize\_32f\_AC4R** (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)

Four channel 32-bit floating-point median filter, ignoring alpha channel.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

#### 7.70.1.42 **NppStatus nppiFilterMedianGetBufferSize\_32f\_C1R** (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)

Single channel 32-bit floating-point median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

#### 7.70.1.43 **NppStatus nppiFilterMedianGetBufferSize\_32f\_C3R** (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)

Three channel 32-bit floating-point median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.44 NppStatus nppiFilterMedianGetBufferSize\_32f\_C4R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)**

Four channel 32-bit floating-point median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.45 NppStatus nppiFilterMedianGetBufferSize\_8u\_AC4R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)**

Four channel 8-bit unsigned median filter, ignoring alpha channel.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.46 NppStatus nppiFilterMedianGetBufferSize\_8u\_C1R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)**

Single channel 8-bit unsigned median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

**7.70.1.47 NppStatus nppiFilterMedianGetBufferSize\_8u\_C3R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)**

Three channel 8-bit unsigned median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

#### 7.70.1.48 NppStatus nppiFilterMedianGetBufferSize\_8u\_C4R (NppiSize *oSizeROI*, NppiSize *oMaskSize*, Npp32u \* *nBufferSize*)

Four channel 8-bit unsigned median filter scratch memory size.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Median operation.

*nBufferSize* Pointer to the size of the scratch buffer required for the Median operation.

**Returns:**

[Image Data Related Error Codes](#)

#### 7.70.1.49 NppStatus nppiFilterMin\_16s\_AC4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)

Four channel 16-bit signed minimum filter, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.50** `NppStatus nppiFilterMin_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 16-bit signed minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.51** `NppStatus nppiFilterMin_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 16-bit signed minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.52** `NppStatus nppiFilterMin_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 16-bit signed minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.53** `NppStatus nppiFilterMin_16u_AC4R (const Npp16u *pSrc, Npp32s nSrcStep, Npp16u *pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 16-bit unsigned minimum filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.54** `NppStatus nppiFilterMin_16u_C1R (const Npp16u *pSrc, Npp32s nSrcStep, Npp16u *pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 16-bit unsigned minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.55** `NppStatus nppiFilterMin_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 16-bit unsigned minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.56** `NppStatus nppiFilterMin_16u_C4R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 16-bit unsigned minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.57** `NppStatus nppiFilterMin_32f_AC4R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 32-bit floating-point minimum filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.58** `NppStatus nppiFilterMin_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Single channel 32-bit floating-point minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.59** `NppStatus nppiFilterMin_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 32-bit floating-point minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.70.1.60 NppStatus nppiFilterMin\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 32-bit floating-point minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.61 NppStatus nppiFilterMin\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Four channel 8-bit unsigned minimum filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.62 NppStatus nppiFilterMin\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiSize *oMaskSize*, NppiPoint *oAnchor*)**

Single channel 8-bit unsigned minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.63** `NppStatus nppiFilterMin_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Three channel 8-bit unsigned minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.70.1.64** `NppStatus nppiFilterMin_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiSize oMaskSize, NppiPoint oAnchor)`

Four channel 8-bit unsigned minimum filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oMaskSize* Width and Height of the neighborhood region for the local Max operation.

*oAnchor* X and Y offsets of the kernel origin frame of reference relative to the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.71 Fixed Filters

Fixed filters perform linear filtering operations (i.e.

### FilterPrewittHoriz

Filters the image using a horizontal Prewitt filter kernel:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterPrewittHoriz_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit unsigned horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit unsigned horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned horizontal Prewitt filter, ignoring alpha channel.*

- `NppStatus nppiFilterPrewittHoriz_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit signed horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed horizontal Prewitt filter, ignoring alpha channel.*

- `NppStatus nppiFilterPrewittHoriz_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 32-bit floating-point horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 32-bit floating-point horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 32-bit floating-point horizontal Prewitt filter.*

- `NppStatus nppiFilterPrewittHoriz_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 32-bit floating-point horizontal Prewitt filter, ignoring alpha channel.*

## FilterPrewittVert

Filters the image using a vertical Prewitt filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterPrewittVert_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit unsigned vertical Prewitt filter.*

- `NppStatus nppiFilterPrewittVert_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit unsigned vertical Prewitt filter.*

- `NppStatus nppiFilterPrewittVert_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned vertical Prewitt filter.*

- `NppStatus nppiFilterPrewittVert_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned vertical Prewitt filter, ignoring alpha channel.*

- `NppStatus nppiFilterPrewittVert_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed vertical Prewitt filter.*

- `NppStatus nppiFilterPrewittVert_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit signed vertical Prewitt filter.*

- `NppStatus nppiFilterPrewittVert_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed vertical Prewitt filter.*

- `NppStatus nppiFilterPrewittVert_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed vertical Prewitt filter, ignoring alpha channel.*

- **NppStatus nppiFilterPrewittVert\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single channel 32-bit floating-point vertical Prewitt filter.*
- **NppStatus nppiFilterPrewittVert\_32f\_C3R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Three channel 32-bit floating-point vertical Prewitt filter.*
- **NppStatus nppiFilterPrewittVert\_32f\_C4R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit floating-point vertical Prewitt filter.*
- **NppStatus nppiFilterPrewittVert\_32f\_AC4R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Four channel 32-bit floating-point vertical Prewitt filter, ignoring alpha channel.*

## FilterScharrHoriz

Filters the image using a horizontal Scharr filter kernel:

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

- **NppStatus nppiFilterScharrHoriz\_8u16s\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter.*
- **NppStatus nppiFilterScharrHoriz\_8s16s\_C1R** (const **Npp8s** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit signed to 16-bit signed horizontal Scharr filter.*
- **NppStatus nppiFilterScharrHoriz\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single channel 32-bit floating-point horizontal Scharr filter.*

## FilterScharrVert

Filters the image using a vertical Scharr filter kernel:

$$\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

- **NppStatus nppiFilterScharrVert\_8u16s\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp16s** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter.*

- `NppStatus nppiFilterScharrVert_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit signed to 16-bit signed vertical Scharr filter.*

- `NppStatus nppiFilterScharrVert_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 32-bit floating-point vertical Scharr filter.*

## FilterScharrHorizBorder

Filters the image using a horizontal Scharr filter kernel with border control:

$$\begin{pmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{pmatrix}$$

- `NppStatus nppiFilterScharrHorizBorder_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter kernel with border control.*

- `NppStatus nppiFilterScharrHorizBorder_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 8-bit signed to 16-bit signed horizontal Scharr filter kernel with border control.*

- `NppStatus nppiFilterScharrHorizBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 32-bit floating-point horizontal Scharr filter kernel with border control.*

## FilterScharrVertBorder

Filters the image using a vertical Scharr filter kernel kernel with border control:

$$\begin{pmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{pmatrix}$$

- `NppStatus nppiFilterScharrVertBorder_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter kernel with border control.*

- `NppStatus nppiFilterScharrVertBorder_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 8-bit signed to 16-bit signed vertical Scharr filter kernel with border control.*

- `NppStatus nppiFilterScharrVertBorder_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `NppiSize` oSrcSize, `NppiPoint` oSrcOffset, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiBorderType` eBorderType)

*Single channel 32-bit floating-point vertical Scharr filter kernel with border control.*

## FilterSobelHoriz

Filters the image using a horizontal Sobel filter kernel:

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 2 & 8 & 12 & 8 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ -1 & -4 & -6 & -4 & -1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelHoriz_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 8-bit unsigned horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHoriz_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 8-bit unsigned horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHoriz_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHoriz_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed horizontal Sobel filter, ignoring alpha channel.*

- `NppStatus nppiFilterSobelHoriz_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Single channel 16-bit signed horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHoriz_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Three channel 16-bit signed horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHoriz_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 16-bit signed horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHoriz_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

*Four channel 8-bit unsigned horizontal Sobel filter, ignoring alpha channel.*

- `NppStatus nppiFilterSobelHoriz_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit floating-point horizontal Sobel filter.*
- `NppStatus nppiFilterSobelHoriz_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit floating-point horizontal Sobel filter.*
- `NppStatus nppiFilterSobelHoriz_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point horizontal Sobel filter.*
- `NppStatus nppiFilterSobelHoriz_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point horizontal Sobel filter, ignoring alpha channel.*
- `NppStatus nppiFilterSobelHoriz_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter.*
- `NppStatus nppiFilterSobelHoriz_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit signed to 16-bit signed horizontal Sobel filter.*
- `NppStatus nppiFilterSobelHorizMask_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 32-bit floating-point horizontal Sobel filter.*

## FilterSobelVert

Filters the image using a vertical Sobel filter kernel:

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & -2 & 0 & 2 & 1 \\ -4 & -8 & 0 & 8 & 4 \\ -6 & -12 & 0 & 12 & 6 \\ -4 & -8 & 0 & 8 & 4 \\ -1 & -2 & 0 & 2 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelVert_8u_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 8-bit unsigned vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_8u_C3R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 8-bit unsigned vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_8u_C4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned vertical Sobel filter.*



- `NppStatus nppiFilterSobelVert_8u_AC4R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp8u` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed vertical Sobel filter; ignoring alpha channel.*
- `NppStatus nppiFilterSobelVert_16s_C1R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 16-bit signed vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_16s_C3R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 16-bit signed vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_16s_C4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 16-bit signed vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 8-bit unsigned vertical Sobel filter; ignoring alpha channel.*
- `NppStatus nppiFilterSobelVert_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Single channel 32-bit floating-point vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_32f_C3R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Three channel 32-bit floating-point vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_32f_C4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_32f_AC4R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)  
*Four channel 32-bit floating-point vertical Sobel filter; ignoring alpha channel.*
- `NppStatus nppiFilterSobelVert_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter.*
- `NppStatus nppiFilterSobelVert_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 8-bit signed to 16-bit signed vertical Sobel filter.*
- `NppStatus nppiFilterSobelVertMask_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)  
*Single channel 32-bit floating-point vertical Sobel filter.*

## FilterSobelHorizSecond

Filters the image using a second derivative, horizontal Sobel filter kernel:

$$\begin{pmatrix} 1 & 2 & 1 \\ -2 & -4 & -2 \\ 1 & 2 & 1 \end{pmatrix} \begin{pmatrix} 1 & 4 & 6 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

- `NppStatus nppiFilterSobelHorizSecond_8u16s_C1R` (const `Npp8u` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHorizSecond_8s16s_C1R` (const `Npp8s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter.*

- `NppStatus nppiFilterSobelHorizSecond_32f_C1R` (const `Npp32f` \*pSrc, `Npp32s` nSrcStep, `Npp32f` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI, `NppiMaskSize` eMaskSize)

*Single channel 32-bit floating-point second derivative, horizontal Sobel filter.*

### 7.71.1 Detailed Description

Fixed filters perform linear filtering operations (i.e. convolutions) with predefined kernels of fixed sizes.

Some of the fixed filters have versions with border control. For these functions, if any portion of the mask overlaps the source image boundary the requested border type operation is applied to all mask pixels which fall outside of the source image.

Currently only the NPP\_BORDER\_REPLICATE border type operation is supported for these functions.

### 7.71.2 Function Documentation

#### 7.71.2.1 `NppStatus nppiFilterPrewittHoriz_16s_AC4R` (const `Npp16s` \*pSrc, `Npp32s` nSrcStep, `Npp16s` \*pDst, `Npp32s` nDstStep, `NppiSize` oSizeROI)

Four channel 16-bit signed horizontal Prewitt filter, ignoring alpha channel.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.2 NppStatus nppiFilterPrewittHoriz\_16s\_C1R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 16-bit signed horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.3 NppStatus nppiFilterPrewittHoriz\_16s\_C3R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Three channel 16-bit signed horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.4 NppStatus nppiFilterPrewittHoriz\_16s\_C4R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Four channel 16-bit signed horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.71.2.5 **NppStatus nppiFilterPrewittHoriz\_32f\_AC4R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 32-bit floating-point horizontal Prewitt filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.71.2.6 **NppStatus nppiFilterPrewittHoriz\_32f\_C1R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

#### 7.71.2.7 **NppStatus nppiFilterPrewittHoriz\_32f\_C3R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 32-bit floating-point horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.8 NppStatus nppiFilterPrewittHoriz\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.9 NppStatus nppiFilterPrewittHoriz\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned horizontal Prewitt filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.10 NppStatus nppiFilterPrewittHoriz\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.11 NppStatus nppiFilterPrewittHoriz\_8u\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 8-bit unsigned horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.12 NppStatus nppiFilterPrewittHoriz\_8u\_C4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned horizontal Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.13 NppStatus nppiFilterPrewittVert\_16s\_AC4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed vertical Prewitt filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.14 NppStatus nppiFilterPrewittVert\_16s\_C1R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 16-bit signed vertical Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.15 NppStatus nppiFilterPrewittVert\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 16-bit signed vertical Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.16 NppStatus nppiFilterPrewittVert\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed vertical Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.17 NppStatus nppiFilterPrewittVert\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point vertical Prewitt filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.18 NppStatus nppiFilterPrewittVert\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 32-bit floating-point vertical Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.19 NppStatus nppiFilterPrewittVert\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 32-bit floating-point vertical Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.71.2.20 NppStatus nppiFilterPrewittVert\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point vertical Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.21 NppStatus nppiFilterPrewittVert\_8u\_AC4R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 8-bit unsigned vertical Prewitt filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.22 NppStatus nppiFilterPrewittVert\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned vertical Prewitt filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.71.2.23 **NppStatus nppiFilterPrewittVert\_8u\_C3R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three channel 8-bit unsigned vertical Prewitt filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.71.2.24 **NppStatus nppiFilterPrewittVert\_8u\_C4R** (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Four channel 8-bit unsigned vertical Prewitt filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.71.2.25 **NppStatus nppiFilterScharrHoriz\_32f\_C1R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single channel 32-bit floating-point horizontal Scharr filter.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.26 NppStatus nppiFilterScharrHoriz\_8s16s\_C1R (const Npp8s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.27 NppStatus nppiFilterScharrHoriz\_8u16s\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.28 NppStatus nppiFilterScharrHorizBorder\_32f\_C1R (const Npp32f \* pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)**

Single channel 32-bit floating-point horizontal Scharr filter kernel with border control.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.29** `NppStatus nppiFilterScharrHorizBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed horizontal Scharr filter kernel with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.30** `NppStatus nppiFilterScharrHorizBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed horizontal Scharr filter kernel with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.31 NppStatus nppiFilterScharrVert\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 32-bit floating-point vertical Scharr filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.32 NppStatus nppiFilterScharrVert\_8s16s\_C1R (const Npp8s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit signed to 16-bit signed vertical Scharr filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.33 NppStatus nppiFilterScharrVert\_8u16s\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.34** `NppStatus nppiFilterScharrVertBorder_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 32-bit floating-point vertical Scharr filter kernel with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.35** `NppStatus nppiFilterScharrVertBorder_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit signed to 16-bit signed vertical Scharr filter kernel with border control.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Source image width and height in pixels relative to pSrc.

*oSrcOffset* Source image starting point relative to pSrc.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.36** `NppStatus nppiFilterScharrVertBorder_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, NppiSize oSrcSize, NppiPoint oSrcOffset, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiBorderType eBorderType)`

Single channel 8-bit unsigned to 16-bit signed vertical Scharr filter kernel with border control.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Source image width and height in pixels relative to pSrc.  
*oSrcOffset* Source image starting point relative to pSrc.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eBorderType* The border type operation to be applied at source image border boundaries.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.37 NppStatus nppiFilterSobelHoriz\_16s\_AC4R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned horizontal Sobel filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.38 NppStatus nppiFilterSobelHoriz\_16s\_C1R (const Npp16s \* pSrc, Npp32s nSrcStep, Npp16s \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 16-bit signed horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.39 NppStatus nppiFilterSobelHoriz\_16s\_C3R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 16-bit signed horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.40 NppStatus nppiFilterSobelHoriz\_16s\_C4R (const Npp16s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 16-bit signed horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.41 NppStatus nppiFilterSobelHoriz\_32f\_AC4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point horizontal Sobel filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.71.2.42 NppStatus nppiFilterSobelHoriz\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single channel 32-bit floating-point horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.43 NppStatus nppiFilterSobelHoriz\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three channel 32-bit floating-point horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.44 NppStatus nppiFilterSobelHoriz\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.45** `NppStatus nppiFilterSobelHoriz_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit signed to 16-bit signed horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.46** `NppStatus nppiFilterSobelHoriz_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit unsigned to 16-bit signed horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.47** `NppStatus nppiFilterSobelHoriz_8u_AC4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed horizontal Sobel filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.48 NppStatus nppiFilterSobelHoriz\_8u\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 8-bit unsigned horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.49 NppStatus nppiFilterSobelHoriz\_8u\_C3R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Three channel 8-bit unsigned horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.50 NppStatus nppiFilterSobelHoriz\_8u\_C4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Four channel 8-bit unsigned horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.51** `NppStatus nppiFilterSobelHorizMask_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.52** `NppStatus nppiFilterSobelHorizSecond_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point second derivative, horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.53** `NppStatus nppiFilterSobelHorizSecond_8s16s_C1R (const Npp8s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit signed to 16-bit signed second derivative, horizontal Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.54** `NppStatus nppiFilterSobelHorizSecond_8u16s_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 8-bit unsigned to 16-bit signed second derivative, horizontal Sobel filter.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.55** `NppStatus nppiFilterSobelVert_16s_AC4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned vertical Sobel filter, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.56** `NppStatus nppiFilterSobelVert_16s_C1R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Single channel 16-bit signed vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.57** `NppStatus nppiFilterSobelVert_16s_C3R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Three channel 16-bit signed vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.58** `NppStatus nppiFilterSobelVert_16s_C4R (const Npp16s * pSrc, Npp32s nSrcStep, Npp16s * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 16-bit signed vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.59 NppStatus nppiFilterSobelVert\_32f\_AC4R (const Npp32f \* pSrc, Npp32s nSrcStep, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Four channel 32-bit floating-point vertical Sobel filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.60 NppStatus nppiFilterSobelVert\_32f\_C1R (const Npp32f \* pSrc, Npp32s nSrcStep, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 32-bit floating-point vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.61 NppStatus nppiFilterSobelVert\_32f\_C3R (const Npp32f \* pSrc, Npp32s nSrcStep, Npp32f \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Three channel 32-bit floating-point vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.62 NppStatus nppiFilterSobelVert\_32f\_C4R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Four channel 32-bit floating-point vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.63 NppStatus nppiFilterSobelVert\_8s16s\_C1R (const Npp8s \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 8-bit signed to 16-bit signed vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.71.2.64 NppStatus nppiFilterSobelVert\_8u16s\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp16s \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*, NppiMaskSize *eMaskSize*)**

Single channel 8-bit unsigned to 16-bit signed vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.71.2.65 NppStatus nppiFilterSobelVert\_8u\_AC4R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Four channel 16-bit signed vertical Sobel filter, ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.66 NppStatus nppiFilterSobelVert\_8u\_C1R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Single channel 8-bit unsigned vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.67 NppStatus nppiFilterSobelVert\_8u\_C3R (const Npp8u \* pSrc, Npp32s nSrcStep, Npp8u \* pDst, Npp32s nDstStep, NppiSize oSizeROI)**

Three channel 8-bit unsigned vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.68** `NppStatus nppiFilterSobelVert_8u_C4R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI)`

Four channel 8-bit unsigned vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.71.2.69** `NppStatus nppiFilterSobelVertMask_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, NppiMaskSize eMaskSize)`

Single channel 32-bit floating-point vertical Sobel filter.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eMaskSize* Enumeration value specifying the mask size.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.72 Geometry Transforms

Routines manipulating an image's geometry.

### Modules

- [ResizeSqrPixel](#)

*ResizeSqrPixel supports the following interpolation modes:.*

- [Resize](#)

*This function has been deprecated.*

- [Remap](#)

*Remap supports the following interpolation modes:.*

- [Rotate](#)

*Rotates an image around the origin (0,0) and then shifts it.*

- [Mirror](#)

- [Affine Transforms](#)

- [Perspective Transform](#)

### 7.72.1 Detailed Description

Routines manipulating an image's geometry.

### 7.72.2 Geometric Transform API Specifics

This section covers some of the unique API features common to the geometric transform primitives.

#### 7.72.2.1 Geometric Transforms and ROIs

Geometric transforms operate on source and destination ROIs. The way these ROIs affect the processing of pixels differs from other (non geometric) image-processing primitives: Only pixels in the intersection of the destination ROI and the transformed source ROI are being processed.

The typical processing proceeds as follows:

1. Transform the rectangular source ROI (given in source image coordinates) into the destination image space. This yields a quadrilateral.
2. Write only pixels in the intersection of the transformed source ROI and the destination ROI.

#### 7.72.2.2 Pixel Interpolation

The majority of image geometry transform operation need to perform a resampling of the source image as source and destination pixels are not coincident.

NPP supports the following pixel interpolation modes (in order from fastest to slowest and lowest to highest quality):

- nearest neighbor
- linear interpolation
- cubic convolution
- supersampling
- interpolation using Lanczos window function

## 7.73 ResizeSqrPixel

ResizeSqrPixel supports the following interpolation modes:.

### GetResizeRect

Returns [NppiRect](#) which represents the offset and size of the destination rectangle that would be generated by resizing the source [NppiRect](#) by the requested scale factors and shifts.

- [NppStatus](#) [nppiGetResizeRect](#) ([NppiRect](#) oSrcROI, [NppiRect](#) \*pDstRect, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)

### ResizeSqrPixel

Resizes images.

- [NppStatus](#) [nppiResizeSqrPixel\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*1 channel 8-bit unsigned image resize.*
- [NppStatus](#) [nppiResizeSqrPixel\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 8-bit unsigned image resize.*
- [NppStatus](#) [nppiResizeSqrPixel\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 8-bit unsigned image resize.*
- [NppStatus](#) [nppiResizeSqrPixel\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 8-bit unsigned image resize not affecting alpha.*
- [NppStatus](#) [nppiResizeSqrPixel\\_8u\\_P3R](#) (const [Npp8u](#) \*const pSrc[3], [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst[3], int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 8-bit unsigned planar image resize.*
- [NppStatus](#) [nppiResizeSqrPixel\\_8u\\_P4R](#) (const [Npp8u](#) \*const pSrc[4], [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp8u](#) \*pDst[4], int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 8-bit unsigned planar image resize.*
- [NppStatus](#) [nppiResizeSqrPixel\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, [NppiSize](#) oSrcSize, int nSrcStep, [NppiRect](#) oSrcROI, [Npp16u](#) \*pDst, int nDstStep, [NppiRect](#) oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*1 channel 16-bit unsigned image resize.*

- `NppStatus nppiResizeSqrPixel_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 16-bit unsigned image resize.*
- `NppStatus nppiResizeSqrPixel_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 16-bit unsigned image resize.*
- `NppStatus nppiResizeSqrPixel_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 16-bit unsigned image resize not affecting alpha.*
- `NppStatus nppiResizeSqrPixel_16u_P3R` (const `Npp16u` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 16-bit unsigned planar image resize.*
- `NppStatus nppiResizeSqrPixel_16u_P4R` (const `Npp16u` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 16-bit unsigned planar image resize.*
- `NppStatus nppiResizeSqrPixel_16s_C1R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*1 channel 16-bit signed image resize.*
- `NppStatus nppiResizeSqrPixel_16s_C3R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 16-bit signed image resize.*
- `NppStatus nppiResizeSqrPixel_16s_C4R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 16-bit signed image resize.*
- `NppStatus nppiResizeSqrPixel_16s_AC4R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 16-bit signed image resize not affecting alpha.*
- `NppStatus nppiResizeSqrPixel_16s_P3R` (const `Npp16s` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16s` \*pDst[3], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 16-bit signed planar image resize.*

- **NppStatus** **nppiResizeSqrPixel\_16s\_P4R** (const **Npp16s** \*const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16s** \*pDst[4], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 16-bit signed planar image resize.*
- **NppStatus** **nppiResizeSqrPixel\_32f\_C1R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*1 channel 32-bit floating point image resize.*
- **NppStatus** **nppiResizeSqrPixel\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 32-bit floating point image resize.*
- **NppStatus** **nppiResizeSqrPixel\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 32-bit floating point image resize.*
- **NppStatus** **nppiResizeSqrPixel\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 32-bit floating point image resize not affecting alpha.*
- **NppStatus** **nppiResizeSqrPixel\_32f\_P3R** (const **Npp32f** \*const pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst[3], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 32-bit floating point planar image resize.*
- **NppStatus** **nppiResizeSqrPixel\_32f\_P4R** (const **Npp32f** \*const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst[4], int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 32-bit floating point planar image resize.*
- **NppStatus** **nppiResizeSqrPixel\_64f\_C1R** (const **Npp64f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*1 channel 64-bit floating point image resize.*
- **NppStatus** **nppiResizeSqrPixel\_64f\_C3R** (const **Npp64f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*3 channel 64-bit floating point image resize.*
- **NppStatus** **nppiResizeSqrPixel\_64f\_C4R** (const **Npp64f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp64f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)  
*4 channel 64-bit floating point image resize.*

- `NppStatus nppiResizeSqrPixel_64f_AC4R` (const `Npp64f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f` \*pDst, int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)

*4 channel 64-bit floating point image resize not affecting alpha.*

- `NppStatus nppiResizeSqrPixel_64f_P3R` (const `Npp64f` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f` \*pDst[3], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)

*3 channel 64-bit floating point planar image resize.*

- `NppStatus nppiResizeSqrPixel_64f_P4R` (const `Npp64f` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f` \*pDst[4], int nDstStep, `NppiRect` oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)

*4 channel 64-bit floating point planar image resize.*

### 7.73.1 Detailed Description

`ResizeSqrPixel` supports the following interpolation modes:.

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
NPPI_INTER_CUBIC2P_BSPLINE
NPPI_INTER_CUBIC2P_CATMULLROM
NPPI_INTER_CUBIC2P_B05C03
NPPI_INTER_SUPER
NPPI_INTER_LANCZOS
```

`ResizeSqrPixel` attempts to choose source pixels that would approximately represent the center of the destination pixels. It does so by using the following scaling formula to select source pixels for interpolation:

```
nAdjustedXFactor = 1.0 / nXFactor;
nAdjustedYFactor = 1.0 / nYFactor;
nAdjustedXShift = nXShift * nAdjustedXFactor + ((1.0 - nAdjustedXFactor) * 0.5);
nAdjustedYShift = nYShift * nAdjustedYFactor + ((1.0 - nAdjustedYFactor) * 0.5);
nSrcX = nAdjustedXFactor * nDstX - nAdjustedXShift;
nSrcY = nAdjustedYFactor * nDstY - nAdjustedYShift;
```

In the `ResizeSqrPixel` functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to `oSizeROI.x` and less than `oSizeROI.x + oSizeROI.width` and greater than or equal to `oSizeROI.y` and less than `oSizeROI.y + oSizeROI.height` then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

### 7.73.2 Error Codes

The resize primitives return the following error codes:

- `NPP_WRONG_INTERSECTION_ROI_ERROR` indicates an error condition if `srcROIRect` has no intersection with the source image.



- [NPP\\_RESIZE\\_NO\\_OPERATION\\_ERROR](#) if either destination ROI width or height is less than 1 pixel.
- [NPP\\_RESIZE\\_FACTOR\\_ERROR](#) Indicates an error condition if either *nXFactor* or *nYFactor* is less than or equal to zero.
- [NPP\\_INTERPOLATION\\_ERROR](#) if *eInterpolation* has an illegal value.
- [NPP\\_SIZE\\_ERROR](#) if source size width or height is less than 2 pixels.

### 7.73.3 Function Documentation

#### 7.73.3.1 **NppStatus nppiGetResizeRect (NppiRect *oSrcROI*, NppiRect \* *pDstRect*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)**

##### Parameters:

- oSrcROI* Region of interest in the source image.
- pDstRect* User supplied host memory pointer to an [NppiRect](#) structure that will be filled in by this function with the region of interest in the destination image.
- nXFactor* Factor by which x dimension is changed.
- nYFactor* Factor by which y dimension is changed.
- nXShift* Source pixel shift in x-direction.
- nYShift* Source pixel shift in y-direction.
- eInterpolation* The type of *eInterpolation* to perform resampling.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

#### 7.73.3.2 **NppStatus nppiResizeSqrPixel\_16s\_AC4R (const Npp16s \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16s \* *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)**

4 channel 16-bit signed image resize not affecting alpha.

##### Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcSize* Size in pixels of the source image.
- oSrcROI* Region of interest in the source image.
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- oDstROI* Region of interest in the destination image.
- nXFactor* Factor by which x dimension is changed.
- nYFactor* Factor by which y dimension is changed.
- nXShift* Source pixel shift in x-direction.
- nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.3 NppStatus nppiResizeSqrPixel\_16s\_C1R (const Npp16s \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s \* pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)**

1 channel 16-bit signed image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.4 NppStatus nppiResizeSqrPixel\_16s\_C3R (const Npp16s \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s \* pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)**

3 channel 16-bit signed image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.5** `NppStatus nppiResizeSqrPixel_16s_C4R (const Npp16s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit signed image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.6** `NppStatus nppiResizeSqrPixel_16s_P3R (const Npp16s *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s * pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 16-bit signed planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pDst*** [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

***nDstStep*** [Destination-Image Line Step](#).

***oDstROI*** Region of interest in the destination image.

***nXFactor*** Factor by which x dimension is changed.

***nYFactor*** Factor by which y dimension is changed.

***nXShift*** Source pixel shift in x-direction.

***nYShift*** Source pixel shift in y-direction.

***eInterpolation*** The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.7** `NppStatus nppiResizeSqrPixel_16s_P4R (const Npp16s *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16s *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit signed planar image resize.

#### Parameters:

***pSrc*** [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSize*** Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pDst*** [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

***nDstStep*** [Destination-Image Line Step](#).

***oDstROI*** Region of interest in the destination image.

***nXFactor*** Factor by which x dimension is changed.

***nYFactor*** Factor by which y dimension is changed.

***nXShift*** Source pixel shift in x-direction.

***nYShift*** Source pixel shift in y-direction.

***eInterpolation*** The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.8** `NppStatus nppiResizeSqrPixel_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit unsigned image resize not affecting alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.9** `NppStatus nppiResizeSqrPixel_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

1 channel 16-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.10** `NppStatus nppiResizeSqrPixel_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 16-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.11** `NppStatus nppiResizeSqrPixel_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.12** `NppStatus nppiResizeSqrPixel_16u_P3R (const Npp16u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 16-bit unsigned planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.13** `NppStatus nppiResizeSqrPixel_16u_P4R (const Npp16u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 16-bit unsigned planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

***nYFactor*** Factor by which y dimension is changed.  
***nXShift*** Source pixel shift in x-direction.  
***nYShift*** Source pixel shift in y-direction.  
***eInterpolation*** The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.14** **NppStatus nppiResizeSqrPixel\_32f\_AC4R** (const Npp32f \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp32f \* *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

4 channel 32-bit floating point image resize not affecting alpha.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).  
***nSrcStep*** [Source-Image Line Step](#).  
***oSrcSize*** Size in pixels of the source image.  
***oSrcROI*** Region of interest in the source image.  
***pDst*** [Destination-Image Pointer](#).  
***nDstStep*** [Destination-Image Line Step](#).  
***oDstROI*** Region of interest in the destination image.  
***nXFactor*** Factor by which x dimension is changed.  
***nYFactor*** Factor by which y dimension is changed.  
***nXShift*** Source pixel shift in x-direction.  
***nYShift*** Source pixel shift in y-direction.  
***eInterpolation*** The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.15** **NppStatus nppiResizeSqrPixel\_32f\_C1R** (const Npp32f \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp32f \* *pDst*, int *nDstStep*, NppiRect *oDstROI*, double *nXFactor*, double *nYFactor*, double *nXShift*, double *nYShift*, int *eInterpolation*)

1 channel 32-bit floating point image resize.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).  
***nSrcStep*** [Source-Image Line Step](#).  
***oSrcSize*** Size in pixels of the source image.  
***oSrcROI*** Region of interest in the source image.  
***pDst*** [Destination-Image Pointer](#).



*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.16** `NppStatus nppiResizeSqrPixel_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 32-bit floating point image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.17** `NppStatus nppiResizeSqrPixel_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 32-bit floating point image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.18** `NppStatus nppiResizeSqrPixel_32f_P3R (const Npp32f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 32-bit floating point planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.19** `NppStatus nppiResizeSqrPixel_32f_P4R (const Npp32f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 32-bit floating point planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.20** `NppStatus nppiResizeSqrPixel_64f_AC4R (const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f *pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 64-bit floating point image resize not affecting alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.21 NppStatus nppiResizeSqrPixel\_64f\_C1R (const Npp64f \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f \* pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)**

1 channel 64-bit floating point image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.22 NppStatus nppiResizeSqrPixel\_64f\_C3R (const Npp64f \* pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f \* pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)**

3 channel 64-bit floating point image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

***nYFactor*** Factor by which y dimension is changed.  
***nXShift*** Source pixel shift in x-direction.  
***nYShift*** Source pixel shift in y-direction.  
***eInterpolation*** The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.23** `NppStatus nppiResizeSqrPixel_64f_C4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 64-bit floating point image resize.

**Parameters:**

***pSrc*** [Source-Image Pointer](#).  
***nSrcStep*** [Source-Image Line Step](#).  
***oSrcSize*** Size in pixels of the source image.  
***oSrcROI*** Region of interest in the source image.  
***pDst*** [Destination-Image Pointer](#).  
***nDstStep*** [Destination-Image Line Step](#).  
***oDstROI*** Region of interest in the destination image.  
***nXFactor*** Factor by which x dimension is changed.  
***nYFactor*** Factor by which y dimension is changed.  
***nXShift*** Source pixel shift in x-direction.  
***nYShift*** Source pixel shift in y-direction.  
***eInterpolation*** The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.24** `NppStatus nppiResizeSqrPixel_64f_P3R (const Npp64f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 64-bit floating point planar image resize.

**Parameters:**

***pSrc*** [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).  
***nSrcStep*** [Source-Image Line Step](#).  
***oSrcSize*** Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pDst*** [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

***nDstStep*** [Destination-Image Line Step](#).

***oDstROI*** Region of interest in the destination image.

***nXFactor*** Factor by which x dimension is changed.

***nYFactor*** Factor by which y dimension is changed.

***nXShift*** Source pixel shift in x-direction.

***nYShift*** Source pixel shift in y-direction.

***eInterpolation*** The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.25** `NppStatus npqiResizeSqrPixel_64f_P4R (const Npp64f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 64-bit floating point planar image resize.

#### Parameters:

***pSrc*** [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSize*** Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pDst*** [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

***nDstStep*** [Destination-Image Line Step](#).

***oDstROI*** Region of interest in the destination image.

***nXFactor*** Factor by which x dimension is changed.

***nYFactor*** Factor by which y dimension is changed.

***nXShift*** Source pixel shift in x-direction.

***nYShift*** Source pixel shift in y-direction.

***eInterpolation*** The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.26** `NppStatus nppiResizeSqrPixel_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 8-bit unsigned image resize not affecting alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.27** `NppStatus nppiResizeSqrPixel_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

1 channel 8-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.28** `NppStatus nppiResizeSqrPixel_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 8-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.29** `NppStatus nppiResizeSqrPixel_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 8-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*nXShift* Source pixel shift in x-direction.  
*nYShift* Source pixel shift in y-direction.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)



**7.73.3.30** `NppStatus nppiResizeSqrPixel_8u_P3R (const Npp8u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst[3], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

3 channel 8-bit unsigned planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.73.3.31** `NppStatus nppiResizeSqrPixel_8u_P4R (const Npp8u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst[4], int nDstStep, NppiRect oDstROI, double nXFactor, double nYFactor, double nXShift, double nYShift, int eInterpolation)`

4 channel 8-bit unsigned planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*nXShift* Source pixel shift in x-direction.

*nYShift* Source pixel shift in y-direction.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

## 7.74 Resize

This function has been deprecated.

### Resize

Resizes images.

- `NppStatus nppiResize_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*1 channel 8-bit unsigned image resize.*
- `NppStatus nppiResize_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*3 channel 8-bit unsigned image resize.*
- `NppStatus nppiResize_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*4 channel 8-bit unsigned image resize.*
- `NppStatus nppiResize_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*4 channel 8-bit unsigned image resize not affecting alpha.*
- `NppStatus nppiResize_8u_P3R` (const `Npp8u` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[3], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*3 channel 8-bit unsigned planar image resize.*
- `NppStatus nppiResize_8u_P4R` (const `Npp8u` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[4], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*4 channel 8-bit unsigned planar image resize.*
- `NppStatus nppiResize_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*1 channel 16-bit unsigned image resize.*
- `NppStatus nppiResize_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)  
*3 channel 16-bit unsigned image resize.*
- `NppStatus nppiResize_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*4 channel 16-bit unsigned image resize.*

- `NppStatus nppiResize_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*4 channel 16-bit unsigned image resize not affecting alpha.*

- `NppStatus nppiResize_16u_P3R` (const `Npp16u` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[3], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*3 channel 16-bit unsigned planar image resize.*

- `NppStatus nppiResize_16u_P4R` (const `Npp16u` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[4], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*4 channel 16-bit unsigned planar image resize.*

- `NppStatus nppiResize_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*1 channel 32-bit floating point image resize.*

- `NppStatus nppiResize_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*3 channel 32-bit floating point image resize.*

- `NppStatus nppiResize_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*4 channel 32-bit floating point image resize.*

- `NppStatus nppiResize_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*4 channel 32-bit floating point image resize not affecting alpha.*

- `NppStatus nppiResize_32f_P3R` (const `Npp32f` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[3], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*3 channel 32-bit floating point planar image resize.*

- `NppStatus nppiResize_32f_P4R` (const `Npp32f` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[4], int nDstStep, `NppiSize` dstROISize, double nXFactor, double nYFactor, int eInterpolation)

*4 channel 32-bit floating point planar image resize.*

### 7.74.1 Detailed Description

This function has been deprecated.

ResizeSqrPixel provides the same functionality and more.

Resize supports the following interpolation modes:

```
NPPI_INTER_NN
NPPI_INTER_LINEAR
NPPI_INTER_CUBIC
NPPI_INTER_SUPER
NPPI_INTER_LANCZOS
```

Resize uses the following scaling formula to select source pixels for interpolation:

```
scaledSrcSize.width = nXFactor * srcRectROI.width;
scaledSrcSize.height = nYFactor * srcRectROI.height;
nAdjustedXFactor = (srcRectROI.width - 1) / (scaledSrcSize.width - 1);
nAdjustedYFactor = (srcRectROI.height - 1) / (scaledSrcSize.height - 1);
nSrcX = nAdjustedXFactor * nDstX;
nSrcY = nAdjustedYFactor * nDstY;
```

In the Resize functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to oSizeROI.x and less than oSizeROI.x + oSizeROI.width and greater than or equal to oSizeROI.y and less than oSizeROI.y + oSizeROI.height then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

## 7.74.2 Error Codes

The resize primitives return the following error codes:

- [NPP\\_WRONG\\_INTERSECTION\\_ROI\\_ERROR](#) indicates an error condition if srcROIRect has no intersection with the source image.
- [NPP\\_RESIZE\\_NO\\_OPERATION\\_ERROR](#) if either destination ROI width or height is less than 1 pixel.
- [NPP\\_RESIZE\\_FACTOR\\_ERROR](#) Indicates an error condition if either nXFactor or nYFactor is less than or equal to zero.
- [NPP\\_INTERPOLATION\\_ERROR](#) if eInterpolation has an illegal value.
- [NPP\\_SIZE\\_ERROR](#) if source size width or height is less than 2 pixels.

## 7.74.3 Function Documentation

**7.74.3.1** `NppStatus nppiResize_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 16-bit unsigned image resize not affecting alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.2** `NppStatus nppiResize_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

1 channel 16-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.3** `NppStatus nppiResize_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 16-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.4** `NppStatus nppiResize_16u_C4R (const Npp16u *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 16-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.5** `NppStatus nppiResize_16u_P3R (const Npp16u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[3], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 16-bit unsigned planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.6** `NppStatus nppiResize_16u_P4R (const Npp16u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u *pDst[4], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 16-bit unsigned planar image resize.

#### Parameters:

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.7** `NppStatus nppiResize_32f_AC4R (const Npp32f *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 32-bit floating point image resize not affecting alpha.

#### Parameters:

*pSrc* [Source-Image Pointer](#).



*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.8** `NppStatus nppiResize_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

1 channel 32-bit floating point image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.9** `NppStatus nppiResize_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 32-bit floating point image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.10** `NppStatus nppiResize_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 32-bit floating point image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.11** `NppStatus nppiResize_32f_P3R (const Npp32f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 32-bit floating point planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).  
*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.12** `NppStatus nppiResize_32f_P4R (const Npp32f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f *pDst[4], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 32-bit floating point planar image resize.

#### Parameters:

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.13** `NppStatus nppiResize_8u_AC4R (const Npp8u *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 8-bit unsigned image resize not affecting alpha.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of interpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.14** `NppStatus nppiResize_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

1 channel 8-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.15** `NppStatus nppiResize_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 8-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.16** `NppStatus nppiResize_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 8-bit unsigned image resize.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*dstROISize* Size in pixels of the destination image.  
*nXFactor* Factor by which x dimension is changed.  
*nYFactor* Factor by which y dimension is changed.  
*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.17** `NppStatus nppiResize_8u_P3R (const Npp8u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

3 channel 8-bit unsigned planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).  
*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.74.3.18** `NppStatus nppiResize_8u_P4R (const Npp8u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiSize dstROISize, double nXFactor, double nYFactor, int eInterpolation)`

4 channel 8-bit unsigned planar image resize.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Planar-Image Pointer Array](#) (host memory array containing device memory image plane pointers).

*nDstStep* [Destination-Image Line Step](#).

*dstROISize* Size in pixels of the destination image.

*nXFactor* Factor by which x dimension is changed.

*nYFactor* Factor by which y dimension is changed.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

## 7.75 Remap

Remap supports the following interpolation modes:.

### Remap

Remaps images.

- `NppStatus nppiRemap_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*1 channel 8-bit unsigned image remap.*
- `NppStatus nppiRemap_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*3 channel 8-bit unsigned image remap.*
- `NppStatus nppiRemap_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*4 channel 8-bit unsigned image remap.*
- `NppStatus nppiRemap_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*4 channel 8-bit unsigned image remap not affecting alpha.*
- `NppStatus nppiRemap_8u_P3R` (const `Npp8u` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp8u` \*pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*3 channel 8-bit unsigned planar image remap.*
- `NppStatus nppiRemap_8u_P4R` (const `Npp8u` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp8u` \*pDst[4], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*4 channel 8-bit unsigned planar image remap.*
- `NppStatus nppiRemap_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*1 channel 16-bit unsigned image remap.*
- `NppStatus nppiRemap_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)  
*3 channel 16-bit unsigned image remap.*
- `NppStatus nppiRemap_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*4 channel 16-bit unsigned image remap.*

- `NppStatus nppiRemap_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*4 channel 16-bit unsigned image remap not affecting alpha.*

- `NppStatus nppiRemap_16u_P3R` (const `Npp16u` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16u` \*pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*3 channel 16-bit unsigned planar image remap.*

- `NppStatus nppiRemap_16u_P4R` (const `Npp16u` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16u` \*pDst[4], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*4 channel 16-bit unsigned planar image remap.*

- `NppStatus nppiRemap_16s_C1R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*1 channel 16-bit signed image remap.*

- `NppStatus nppiRemap_16s_C3R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*3 channel 16-bit signed image remap.*

- `NppStatus nppiRemap_16s_C4R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*4 channel 16-bit signed image remap.*

- `NppStatus nppiRemap_16s_AC4R` (const `Npp16s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*4 channel 16-bit signed image remap not affecting alpha.*

- `NppStatus nppiRemap_16s_P3R` (const `Npp16s` \*const pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16s` \*pDst[3], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*3 channel 16-bit signed planar image remap.*

- `NppStatus nppiRemap_16s_P4R` (const `Npp16s` \*const pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp16s` \*pDst[4], int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*4 channel 16-bit signed planar image remap.*

- `NppStatus nppiRemap_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const `Npp32f` \*pXMap, int nXMapStep, const `Npp32f` \*pYMap, int nYMapStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oDstSizeROI, int eInterpolation)

*1 channel 32-bit floating point image remap.*



- **NppStatus nppiRemap\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp32f** \*pXMap, int nXMapStep, const **Npp32f** \*pYMap, int nYMapStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*3 channel 32-bit floating point image remap.*
- **NppStatus nppiRemap\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp32f** \*pXMap, int nXMapStep, const **Npp32f** \*pYMap, int nYMapStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*4 channel 32-bit floating point image remap.*
- **NppStatus nppiRemap\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp32f** \*pXMap, int nXMapStep, const **Npp32f** \*pYMap, int nYMapStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*4 channel 32-bit floating point image remap not affecting alpha.*
- **NppStatus nppiRemap\_32f\_P3R** (const **Npp32f** \*const pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp32f** \*pXMap, int nXMapStep, const **Npp32f** \*pYMap, int nYMapStep, **Npp32f** \*pDst[3], int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*3 channel 32-bit floating point planar image remap.*
- **NppStatus nppiRemap\_32f\_P4R** (const **Npp32f** \*const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp32f** \*pXMap, int nXMapStep, const **Npp32f** \*pYMap, int nYMapStep, **Npp32f** \*pDst[4], int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*4 channel 32-bit floating point planar image remap.*
- **NppStatus nppiRemap\_64f\_C1R** (const **Npp64f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp64f** \*pXMap, int nXMapStep, const **Npp64f** \*pYMap, int nYMapStep, **Npp64f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*1 channel 64-bit floating point image remap.*
- **NppStatus nppiRemap\_64f\_C3R** (const **Npp64f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp64f** \*pXMap, int nXMapStep, const **Npp64f** \*pYMap, int nYMapStep, **Npp64f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*3 channel 64-bit floating point image remap.*
- **NppStatus nppiRemap\_64f\_C4R** (const **Npp64f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp64f** \*pXMap, int nXMapStep, const **Npp64f** \*pYMap, int nYMapStep, **Npp64f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*4 channel 64-bit floating point image remap.*
- **NppStatus nppiRemap\_64f\_AC4R** (const **Npp64f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp64f** \*pXMap, int nXMapStep, const **Npp64f** \*pYMap, int nYMapStep, **Npp64f** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*4 channel 64-bit floating point image remap not affecting alpha.*
- **NppStatus nppiRemap\_64f\_P3R** (const **Npp64f** \*const pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp64f** \*pXMap, int nXMapStep, const **Npp64f** \*pYMap, int nYMapStep, **Npp64f** \*pDst[3], int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)  
*3 channel 64-bit floating point planar image remap.*

- **NppStatus nppiRemap\_64f\_P4R** (const **Npp64f** \*const pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp64f** \*pXMap, int nXMapStep, const **Npp64f** \*pYMap, int nYMapStep, **Npp64f** \*pDst[4], int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)

*4 channel 64-bit floating point planar image remap.*

### 7.75.1 Detailed Description

Remap supports the following interpolation modes:.

NPPI\_INTER\_NN NPPI\_INTER\_LINEAR NPPI\_INTER\_CUBIC NPPI\_INTER\_CUBIC2P\_BSPLINE  
NPPI\_INTER\_CUBIC2P\_CATMULLROM NPPI\_INTER\_CUBIC2P\_B05C03 NPPI\_INTER\_-  
LANCZOS

Remap chooses source pixels using pixel coordinates explicitly supplied in two 2D device memory image arrays pointed to by the pXMap and pYMap pointers. The pXMap array contains the X coordinated and the pYMap array contains the Y coordinate of the corresponding source image pixel to use as input. These coordinates are in floating point format so fraction pixel positions can be used. The coordinates of the source pixel to sample are determined as follows:

$nSrcX = pXMap[nDstX, nDstY]$   $nSrcY = pYMap[nDstX, nDstY]$

In the Remap functions below source image clip checking is handled as follows:

If the source pixel fractional x and y coordinates are greater than or equal to oSizeROI.x and less than oSizeROI.x + oSizeROI.width and greater than or equal to oSizeROI.y and less than oSizeROI.y + oSizeROI.height then the source pixel is considered to be within the source image clip rectangle and the source image is sampled. Otherwise the source image is not sampled and a destination pixel is not written to the destination image.

### 7.75.2 Error Codes

The remap primitives return the following error codes:

- **NPP\_WRONG\_INTERSECTION\_ROI\_ERROR** indicates an error condition if srcROIRect has no intersection with the source image.
- **NPP\_INTERPOLATION\_ERROR** if eInterpolation has an illegal value.

### 7.75.3 Function Documentation

**7.75.3.1 NppStatus nppiRemap\_16s\_AC4R** (const **Npp16s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const **Npp32f** \*pXMap, int nXMapStep, const **Npp32f** \*pYMap, int nYMapStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oDstSizeROI, int eInterpolation)

4 channel 16-bit signed image remap not affecting alpha.

**Parameters:**

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**oSrcSize** Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of interpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.2** `NppStatus nppiRemap_16s_C1R (const Npp16s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16s * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 16-bit signed image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

### 7.75.3.3 NppStatus nppiRemap\_16s\_C3R (const Npp16s \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, const Npp32f \* *pXMap*, int *nXMapStep*, const Npp32f \* *pYMap*, int *nYMapStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *eInterpolation*)

3 channel 16-bit signed image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of *eInterpolation* to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

### 7.75.3.4 NppStatus nppiRemap\_16s\_C4R (const Npp16s \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, const Npp32f \* *pXMap*, int *nXMapStep*, const Npp32f \* *pYMap*, int *nYMapStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *eInterpolation*)

4 channel 16-bit signed image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.5** `NppStatus nppiRemap_16s_P3R (const Npp16s *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp16s *pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 16-bit signed planar image remap.

#### Parameters:

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Planar-Image Pointer Array](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.6** `NppStatus nppiRemap_16s_P4R (const Npp16s *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp16s *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 16-bit signed planar image remap.

#### Parameters:

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Planar-Image Pointer Array](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.7** `NppStatus nppiRemap_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 16-bit unsigned image remap not affecting alpha.

#### Parameters:

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSize*** Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of interpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

### 7.75.3.8 NppStatus nppiRemap\_16u\_C1R (const Npp16u \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, const Npp32f \* *pXMap*, int *nXMapStep*, const Npp32f \* *pYMap*, int *nYMapStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *eInterpolation*)

1 channel 16-bit unsigned image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of *eInterpolation* to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

### 7.75.3.9 NppStatus nppiRemap\_16u\_C3R (const Npp16u \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, const Npp32f \* *pXMap*, int *nXMapStep*, const Npp32f \* *pYMap*, int *nYMapStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oDstSizeROI*, int *eInterpolation*)

3 channel 16-bit unsigned image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.10** `NppStatus nppiRemap_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 16-bit unsigned image remap.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.11** `NppStatus nppiRemap_16u_P3R (const Npp16u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp16u * pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 16-bit unsigned planar image remap.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.



***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Planar-Image Pointer Array](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of *eInterpolation* to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.12** `NppStatus nppiRemap_16u_P4R (const Npp16u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp16u *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 16-bit unsigned planar image remap.

#### Parameters:

***pSrc*** [Source-Planar-Image Pointer Array](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSize*** Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Planar-Image Pointer Array](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of *eInterpolation* to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.13** `NppStatus nppiRemap_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 32-bit floating point image remap not affecting alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.14** `NppStatus nppiRemap_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 32-bit floating point image remap.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.15** `NppStatus nppiRemap_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 32-bit floating point image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.16** `NppStatus nppiRemap_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp32f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 32-bit floating point image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of *eInterpolation* to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.17** `NppStatus nppiRemap_32f_P3R (const Npp32f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp32f *pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 32-bit floating point planar image remap.

#### Parameters:

***pSrc*** [Source-Planar-Image Pointer Array](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSize*** Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Planar-Image Pointer Array](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of *eInterpolation* to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.18** `NppStatus nppiRemap_32f_P4R (const Npp32f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp32f *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 32-bit floating point planar image remap.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.

*pDst* [Destination-Planar-Image Pointer Array](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of *eInterpolation* to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.19** `NppStatus nppiRemap_64f_AC4R (const Npp64f *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep, const Npp64f *pYMap, int nYMapStep, Npp64f *pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 64-bit floating point image remap not affecting alpha.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.

*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Region of interest size in the destination image.  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.20** `NppStatus nppiRemap_64f_C1R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f * pXMap, int nXMapStep, const Npp64f * pYMap, int nYMapStep, Npp64f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 64-bit floating point image remap.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.  
*oSrcROI* Region of interest in the source image.  
*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.  
*nXMapStep* pXMap image array line step in bytes.  
*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.  
*nYMapStep* pYMap image array line step in bytes.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstSizeROI* Region of interest size in the destination image.  
*eInterpolation* The type of eInterpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.21** `NppStatus nppiRemap_64f_C3R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f * pXMap, int nXMapStep, const Npp64f * pYMap, int nYMapStep, Npp64f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 64-bit floating point image remap.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.22** `NppStatus nppiRemap_64f_C4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f * pXMap, int nXMapStep, const Npp64f * pYMap, int nYMapStep, Npp64f * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 64-bit floating point image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.23** `NppStatus nppiRemap_64f_P3R (const Npp64f *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep, const Npp64f *pYMap, int nYMapStep, Npp64f *pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 64-bit floating point planar image remap.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.

*pDst* [Destination-Planar-Image Pointer Array](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of *eInterpolation* to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.24** `NppStatus nppiRemap_64f_P4R (const Npp64f *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp64f *pXMap, int nXMapStep, const Npp64f *pYMap, int nYMapStep, Npp64f *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 64-bit floating point planar image remap.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* *pXMap* image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* *pYMap* image array line step in bytes.



*pDst* [Destination-Planar-Image Pointer Array](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.25** `NppStatus nppiRemap_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 8-bit unsigned image remap not affecting alpha.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of interpolation to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.26** `NppStatus nppiRemap_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

1 channel 8-bit unsigned image remap.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of *eInterpolation* to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.27** `NppStatus nppiRemap_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 8-bit unsigned image remap.

#### Parameters:

***pSrc*** [Source-Image Pointer](#).

***nSrcStep*** [Source-Image Line Step](#).

***oSrcSize*** Size in pixels of the source image.

***oSrcROI*** Region of interest in the source image.

***pXMap*** Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

***nXMapStep*** *pXMap* image array line step in bytes.

***pYMap*** Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

***nYMapStep*** *pYMap* image array line step in bytes.

***pDst*** [Destination-Image Pointer](#).

***nDstStep*** [Destination-Image Line Step](#).

***oDstSizeROI*** Region of interest size in the destination image.

***eInterpolation*** The type of *eInterpolation* to perform resampling.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.28** `NppStatus nppiRemap_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst, int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 8-bit unsigned image remap.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.29** `NppStatus nppiRemap_8u_P3R (const Npp8u *const pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f * pXMap, int nXMapStep, const Npp32f * pYMap, int nYMapStep, Npp8u * pDst[3], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

3 channel 8-bit unsigned planar image remap.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Planar-Image Pointer Array](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

**7.75.3.30** `NppStatus nppiRemap_8u_P4R (const Npp8u *const pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const Npp32f *pXMap, int nXMapStep, const Npp32f *pYMap, int nYMapStep, Npp8u *pDst[4], int nDstStep, NppiSize oDstSizeROI, int eInterpolation)`

4 channel 8-bit unsigned planar image remap.

**Parameters:**

*pSrc* [Source-Planar-Image Pointer Array](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image.

*oSrcROI* Region of interest in the source image.

*pXMap* Device memory pointer to 2D image array of X coordinate values to be used when sampling source image.

*nXMapStep* pXMap image array line step in bytes.

*pYMap* Device memory pointer to 2D image array of Y coordinate values to be used when sampling source image.

*nYMapStep* pYMap image array line step in bytes.

*pDst* [Destination-Planar-Image Pointer Array](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstSizeROI* Region of interest size in the destination image.

*eInterpolation* The type of eInterpolation to perform resampling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Error Codes](#)

## 7.76 Rotate

Rotates an image around the origin (0,0) and then shifts it.

### Utility Functions

- **NppStatus** **nppiGetRotateQuad** (**NppiRect** oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)  
*Compute shape of rotated image.*
- **NppStatus** **nppiGetRotateBound** (**NppiRect** oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)  
*Compute bounding-box of rotated image.*

### Rotate

- **NppStatus** **nppiRotate\_8u\_C1R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*8-bit unsigned image rotate.*
- **NppStatus** **nppiRotate\_8u\_C3R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*3 channel 8-bit unsigned image rotate.*
- **NppStatus** **nppiRotate\_8u\_C4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*4 channel 8-bit unsigned image rotate.*
- **NppStatus** **nppiRotate\_8u\_AC4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*4 channel 8-bit unsigned image rotate ignoring alpha channel.*
- **NppStatus** **nppiRotate\_16u\_C1R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*16-bit unsigned image rotate.*
- **NppStatus** **nppiRotate\_16u\_C3R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)  
*3 channel 16-bit unsigned image rotate.*
- **NppStatus** **nppiRotate\_16u\_C4R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 16-bit unsigned image rotate.*

- **NppStatus nppiRotate\_16u\_AC4R** (const **Npp16u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp16u** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 16-bit unsigned image rotate ignoring alpha channel.*

- **NppStatus nppiRotate\_32f\_C1R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*32-bit float image rotate.*

- **NppStatus nppiRotate\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*3 channel 32-bit float image rotate.*

- **NppStatus nppiRotate\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 32-bit float image rotate.*

- **NppStatus nppiRotate\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)

*4 channel 32-bit float image rotate ignoring alpha channel.*

### 7.76.1 Detailed Description

Rotates an image around the origin (0,0) and then shifts it.

### 7.76.2 Rotate Error Codes

- **NPP\_INTERPOLATION\_ERROR** if eInterpolation has an illegal value.
- **NPP\_RECTANGLE\_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1.
- **NPP\_WRONG\_INTERSECTION\_ROI\_ERROR** indicates an error condition if srcROIrect has no intersection with the source image.
- **NPP\_WRONG\_INTERSECTION\_QUAD\_WARNING** indicates a warning that no operation is performed if the transformed source ROI does not intersect the destination ROI.

### 7.76.3 Function Documentation

#### 7.76.3.1 NppStatus nppiGetRotateBound (NppiRect oSrcROI, double aBoundingBox[2][2], double nAngle, double nShiftX, double nShiftY)

Compute bounding-box of rotated image.

**Parameters:**

*oSrcROI* Region-of-interest of the source image.

*aBoundingBox* Two 2D points representing the bounding-box of the rotated image. All four points from `nppiGetRotateQuad` are contained inside the axis-aligned rectangle spanned by the two points of this bounding box.

*nAngle* The rotation angle.

*nShiftX* Post-rotation shift in x-direction.

*nShiftY* Post-rotation shift in y-direction.

**Returns:**

[ROI Related Error Codes.](#)

### 7.76.3.2 `NppStatus nppiGetRotateQuad (NppiRect oSrcROI, double aQuad[4][2], double nAngle, double nShiftX, double nShiftY)`

Compute shape of rotated image.

**Parameters:**

*oSrcROI* Region-of-interest of the source image.

*aQuad* Array of 2D points. These points are the locations of the corners of the rotated ROI.

*nAngle* The rotation nAngle.

*nShiftX* Post-rotation shift in x-direction

*nShiftY* Post-rotation shift in y-direction

**Returns:**

[ROI Related Error Codes.](#)

### 7.76.3.3 `NppStatus nppiRotate_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 16-bit unsigned image rotate ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer.](#)

*nSrcStep* [Source-Image Line Step.](#)

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer.](#)

*nDstStep* [Destination-Image Line Step.](#)

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.4** `NppStatus nppiRotate_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

16-bit unsigned image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.5** `NppStatus nppiRotate_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 16-bit unsigned image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.



*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.6** `NppStatus nppiRotate_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 16-bit unsigned image rotate.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.

*nShiftX* Shift along horizontal axis

*nShiftY* Shift along vertical axis

*eInterpolation* The type of interpolation to perform resampling

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.7** `NppStatus nppiRotate_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 32-bit float image rotate ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcSize* Size in pixels of the source image

*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.

*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.8** `NppStatus nppiRotate_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

32-bit float image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.9** `NppStatus nppiRotate_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 32-bit float image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.10** `NppStatus nppiRotate_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 32-bit float image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.11** `NppStatus nppiRotate_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 8-bit unsigned image rotate ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.12** `NppStatus nppiRotate_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

8-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.13** `NppStatus nppiRotate_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

3 channel 8-bit unsigned image rotate.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.

*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

**7.76.3.14** `NppStatus nppiRotate_8u_C4R (const Npp8u *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u *pDst, int nDstStep, NppiRect oDstROI, double nAngle, double nShiftX, double nShiftY, int eInterpolation)`

4 channel 8-bit unsigned image rotate.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcSize* Size in pixels of the source image  
*oSrcROI* Region of interest in the source image.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Region of interest in the destination image.  
*nAngle* The angle of rotation in degrees.  
*nShiftX* Shift along horizontal axis  
*nShiftY* Shift along vertical axis  
*eInterpolation* The type of interpolation to perform resampling

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Rotate Error Codes](#)

## 7.77 Mirror

### Mirror

Mirrors images horizontally, vertically and diagonally.

- **NppStatus nppiMirror\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*1 channel 8-bit unsigned image mirror.*
- **NppStatus nppiMirror\_8u\_C1IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*1 channel 8-bit unsigned in place image mirror.*
- **NppStatus nppiMirror\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*3 channel 8-bit unsigned image mirror.*
- **NppStatus nppiMirror\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*3 channel 8-bit unsigned in place image mirror.*
- **NppStatus nppiMirror\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 8-bit unsigned image mirror.*
- **NppStatus nppiMirror\_8u\_C4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 8-bit unsigned in place image mirror.*
- **NppStatus nppiMirror\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 8-bit unsigned image mirror not affecting alpha.*
- **NppStatus nppiMirror\_8u\_AC4IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 8-bit unsigned in place image mirror not affecting alpha.*
- **NppStatus nppiMirror\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*1 channel 16-bit unsigned image mirror.*
- **NppStatus nppiMirror\_16u\_C1IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*1 channel 16-bit unsigned in place image mirror.*
- **NppStatus nppiMirror\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*3 channel 16-bit unsigned image mirror.*

- **NppStatus** **nppiMirror\_16u\_C3IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*3 channel 16-bit unsigned in place image mirror.*
- **NppStatus** **nppiMirror\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit unsigned image mirror.*
- **NppStatus** **nppiMirror\_16u\_C4IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit unsigned in place image mirror.*
- **NppStatus** **nppiMirror\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit unsigned image mirror not affecting alpha.*
- **NppStatus** **nppiMirror\_16u\_AC4IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit unsigned in place image mirror not affecting alpha.*
- **NppStatus** **nppiMirror\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*1 channel 16-bit signed image mirror.*
- **NppStatus** **nppiMirror\_16s\_C1IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*1 channel 16-bit signed in place image mirror.*
- **NppStatus** **nppiMirror\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*3 channel 16-bit signed image mirror.*
- **NppStatus** **nppiMirror\_16s\_C3IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*3 channel 16-bit signed in place image mirror.*
- **NppStatus** **nppiMirror\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit signed image mirror.*
- **NppStatus** **nppiMirror\_16s\_C4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit signed in place image mirror.*
- **NppStatus** **nppiMirror\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit signed image mirror not affecting alpha.*
- **NppStatus** **nppiMirror\_16s\_AC4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oROI, **NppiAxis** flip)  
*4 channel 16-bit signed in place image mirror not affecting alpha.*

- `NppStatus nppiMirror_32s_C1R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*1 channel 32-bit image mirror.*
- `NppStatus nppiMirror_32s_C1IR` (`Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*1 channel 32-bit signed in place image mirror.*
- `NppStatus nppiMirror_32s_C3R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 32-bit image mirror.*
- `NppStatus nppiMirror_32s_C3IR` (`Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 32-bit signed in place image mirror.*
- `NppStatus nppiMirror_32s_C4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 32-bit image mirror.*
- `NppStatus nppiMirror_32s_C4IR` (`Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 32-bit signed in place image mirror.*
- `NppStatus nppiMirror_32s_AC4R` (const `Npp32s` \*pSrc, int nSrcStep, `Npp32s` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 32-bit image mirror not affecting alpha.*
- `NppStatus nppiMirror_32s_AC4IR` (`Npp32s` \*pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*4 channel 32-bit signed in place image mirror not affecting alpha.*
- `NppStatus nppiMirror_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*1 channel 32-bit float image mirror.*
- `NppStatus nppiMirror_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*1 channel 32-bit float in place image mirror.*
- `NppStatus nppiMirror_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 32-bit float image mirror.*
- `NppStatus nppiMirror_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oROI, `NppiAxis` flip)  
*3 channel 32-bit float in place image mirror.*
- `NppStatus nppiMirror_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oROI, `NppiAxis` flip)



*4 channel 32-bit float image mirror.*

- **NppStatus** [nppiMirror\\_32f\\_C4IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oROI, [NppiAxis](#) flip)

*4 channel 32-bit float in place image mirror.*

- **NppStatus** [nppiMirror\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oROI, [NppiAxis](#) flip)

*4 channel 32-bit float image mirror not affecting alpha.*

- **NppStatus** [nppiMirror\\_32f\\_AC4IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oROI, [NppiAxis](#) flip)

*4 channel 32-bit float in place image mirror not affecting alpha.*

### 7.77.1 Detailed Description

### 7.77.2 Mirror Error Codes

- `NPP_MIRROR_FLIP_ERR` if flip has an illegal value.

### 7.77.3 Function Documentation

#### 7.77.3.1 **NppStatus** [nppiMirror\\_16s\\_AC4IR](#) ([Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oROI, [NppiAxis](#) flip)

*4 channel 16-bit signed in place image mirror not affecting alpha.*

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

#### 7.77.3.2 **NppStatus** [nppiMirror\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oROI, [NppiAxis](#) flip)

*4 channel 16-bit signed image mirror not affecting alpha.*

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* [Region-of-Interest \(ROI\)](#).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.3 NppStatus nppiMirror\_16s\_C1IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)**

1 channel 16-bit signed in place image mirror.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.4 NppStatus nppiMirror\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

1 channel 16-bit signed image mirror.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oROI* [Region-of-Interest \(ROI\)](#).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.5 NppStatus nppiMirror\_16s\_C3IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 16-bit signed in place image mirror.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

#### 7.77.3.6 NppStatus nppiMirror\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 16-bit signed image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

#### 7.77.3.7 NppStatus nppiMirror\_16s\_C4IR (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 16-bit signed in place image mirror.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

#### 7.77.3.8 NppStatus nppiMirror\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 16-bit signed image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

**7.77.3.9 NppStatus nppiMirror\_16u\_AC4IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 16-bit unsigned in place image mirror not affecting alpha.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

**7.77.3.10 NppStatus nppiMirror\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

4 channel 16-bit unsigned image mirror not affecting alpha.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Mirror Error Codes

**7.77.3.11 NppStatus nppiMirror\_16u\_C1IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)**

1 channel 16-bit unsigned in place image mirror.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.12 NppStatus nppiMirror\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oROI, NppiAxis flip)**

1 channel 16-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.13 NppStatus nppiMirror\_16u\_C3IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oROI, NppiAxis flip)**

3 channel 16-bit unsigned in place image mirror.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.14 NppStatus nppiMirror\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 16-bit unsigned image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.15 NppStatus nppiMirror\_16u\_C4IR (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 16-bit unsigned in place image mirror.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.16 NppStatus nppiMirror\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 16-bit unsigned image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.17 NppStatus nppiMirror\_32f\_AC4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float in place image mirror not affecting alpha.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.18 NppStatus nppiMirror\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float image mirror not affecting alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.19 NppStatus nppiMirror\_32f\_C1IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit float in place image mirror.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.20 **NppStatus nppiMirror\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit float image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.21 **NppStatus nppiMirror\_32f\_C3IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 32-bit float in place image mirror.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.22 **NppStatus nppiMirror\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 32-bit float image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)



### 7.77.3.23 NppStatus nppiMirror\_32f\_C4IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float in place image mirror.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.24 NppStatus nppiMirror\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit float image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.25 NppStatus nppiMirror\_32s\_AC4IR (Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit signed in place image mirror not affecting alpha.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.26 **NppStatus nppiMirror\_32s\_AC4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit image mirror not affecting alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.27 **NppStatus nppiMirror\_32s\_C11R** (Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit signed in place image mirror.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.28 **NppStatus nppiMirror\_32s\_C1R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

1 channel 32-bit image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.29 NppStatus nppiMirror\_32s\_C3IR (Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)**

3 channel 32-bit signed in place image mirror.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.30 NppStatus nppiMirror\_32s\_C3R (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)**

3 channel 32-bit image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.31 NppStatus nppiMirror\_32s\_C4IR (Npp32s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)**

4 channel 32-bit signed in place image mirror.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.32 **NppStatus nppiMirror\_32s\_C4R** (const Npp32s \* *pSrc*, int *nSrcStep*, Npp32s \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 32-bit image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.33 **NppStatus nppiMirror\_8u\_AC4IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned in place image mirror not affecting alpha.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.34 **NppStatus nppiMirror\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned image mirror not affecting alpha.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.35 NppStatus nppiMirror\_8u\_C1IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)**

1 channel 8-bit unsigned in place image mirror.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.36 NppStatus nppiMirror\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)**

1 channel 8-bit unsigned image mirror.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

**7.77.3.37 NppStatus nppiMirror\_8u\_C3IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)**

3 channel 8-bit unsigned in place image mirror.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.38 **NppStatus nppiMirror\_8u\_C3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

3 channel 8-bit unsigned image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.39 **NppStatus nppiMirror\_8u\_C4IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned in place image mirror.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

### 7.77.3.40 **NppStatus nppiMirror\_8u\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oROI*, NppiAxis *flip*)

4 channel 8-bit unsigned image mirror.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Distance in bytes between starts of consecutive lines of the destination image.

*oROI* Region-of-Interest (ROI).

*flip* Specifies the axis about which the image is to be mirrored.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Mirror Error Codes](#)

## 7.78 Affine Transforms

### Utility Functions

- **NppStatus** **nppiGetAffineTransform** (**NppiRect** oSrcROI, const double aQuad[4][2], double aCoeffs[2][3])

*Computes affine transform coefficients based on source ROI and destination quadrilateral.*

- **NppStatus** **nppiGetAffineQuad** (**NppiRect** oSrcROI, double aQuad[4][2], const double aCoeffs[2][3])

*Compute shape of transformed image.*

- **NppStatus** **nppiGetAffineBound** (**NppiRect** oSrcROI, double aBound[2][2], const double aCoeffs[2][3])

*Compute bounding-box of transformed image.*

### Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a  $2 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates are computed as follows:

$$x' = c_{00} * x + c_{01} * y + c_{02} \quad y' = c_{10} * x + c_{11} * y + c_{12} \quad C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}$$

Affine transforms can be understood as a linear transformation (traditional matrix multiplication) and a shift operation. The  $2 \times 2$  matrix

$$L = \begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}$$

represents the linear transform portion of the affine transformation. The vector

$$v = \begin{pmatrix} c_{02} \\ c_{12} \end{pmatrix}$$

represents the post-transform shift, i.e. after the pixel location is transformed by  $L$  it is translated by  $v$ .

- **NppStatus** **nppiWarpAffine\_8u\_C1R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Single-channel 8-bit unsigned affine warp.*

- **NppStatus** **nppiWarpAffine\_8u\_C3R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 8-bit unsigned affine warp.*

- **NppStatus** **nppiWarpAffine\_8u\_C4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 8-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_8u_AC4R` (const `Npp8u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 8-bit unsigned affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffine_8u_P3R` (const `Npp8u *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel planar 8-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_8u_P4R` (const `Npp8u *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp8u *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel planar 8-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_16u_C1R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Single-channel 16-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_16u_C3R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel 16-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_16u_C4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 16-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_16u_AC4R` (const `Npp16u *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel 16-bit unsigned affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffine_16u_P3R` (const `Npp16u *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Three-channel planar 16-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_16u_P4R` (const `Npp16u *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp16u *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Four-channel planar 16-bit unsigned affine warp.*

- `NppStatus nppiWarpAffine_32s_C1R` (const `Npp32s *pSrc`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, `Npp32s *pDst`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aCoeffs[2][3]`, int `eInterpolation`)

*Single-channel 32-bit signed affine warp.*



- `NppStatus nppiWarpAffine_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 32-bit signed affine warp.*

- `NppStatus nppiWarpAffine_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit signed affine warp.*

- `NppStatus nppiWarpAffine_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit signed affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffine_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel planar 32-bit signed affine warp.*

- `NppStatus nppiWarpAffine_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel planar 32-bit signed affine warp.*

- `NppStatus nppiWarpAffine_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Single-channel 32-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 32-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit floating-point affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffine_32f_P3R` (const `Npp32f` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel planar 32-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f *pDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel planar 32-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_64f_C1R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Single-channel 64-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_64f_C3R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 64-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_64f_C4R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 64-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_64f_AC4R` (const `Npp64f *pSrc`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *pDst`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 64-bit floating-point affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffine_64f_P3R` (const `Npp64f *aSrc[3]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *aDst[3]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel planar 64-bit floating-point affine warp.*

- `NppStatus nppiWarpAffine_64f_P4R` (const `Npp64f *aSrc[4]`, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp64f *aDst[4]`, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel planar 64-bit floating-point affine warp.*

## Backwards Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is given as a  $2 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates fulfill the following properties:

$$x = c_{00} * x' + c_{01} * y' + c_{02} \quad y = c_{10} * x' + c_{11} * y' + c_{12} \quad C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \end{bmatrix}$$

In other words, given matrix  $C$  the source image's shape is transformed to the destination image using the inverse matrix  $C^{-1}$ :

$$M = C^{-1} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \end{bmatrix} \quad x' = m_{00} * x + m_{01} * y + m_{02} \quad y' = m_{10} * x + m_{11} * y + m_{12}$$

- `NppStatus nppiWarpAffineBack_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Single-channel 8-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 8-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 8-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 8-bit unsigned integer backwards affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineBack_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel planar 8-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel planar 8-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Single-channel 16-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 16-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 16-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 16-bit unsigned integer backwards affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineBack_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel planar 16-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel planar 16-bit unsigned integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Single-channel 32-bit signed integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 32-bit signed integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit signed integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit signed integer backwards affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineBack_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel planar 32-bit signed integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel planar 32-bit signed integer backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Single-channel 32-bit floating-point backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel 32-bit floating-point backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit floating-point backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel 32-bit floating-point backwards affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineBack_32f_P3R` (const `Npp32f` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Three-channel planar 32-bit floating-point backwards affine warp.*

- `NppStatus nppiWarpAffineBack_32f_P4R` (const `Npp32f` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[2][3], int eInterpolation)

*Four-channel planar 32-bit floating-point backwards affine warp.*

## Quad-Based Affine Transform

Transforms (warps) an image based on an affine transform.

The affine transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

An affine transform is fully determined by the mapping of 3 discrete points. The following primitives compute an affine transformation matrix that maps the first three corners of the source quad are mapped to the first three vertices of the destination image quad. If the fourth vertices do not match the transform, an `NPP_AFFINE_QUAD_INCORRECT_WARNING` is returned by the primitive.

- `NppStatus nppiWarpAffineQuad_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 32-bit floating-point quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 8-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 8-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 8-bit unsigned integer quad-based affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineQuad_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 8-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 8-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 16-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 16-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 16-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 16-bit unsigned integer quad-based affine warp, ignoring alpha channel.*

- `NppStatus nppiWarpAffineQuad_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 16-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 16-bit unsigned integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 32-bit signed integer quad-based affine warp.*

- `NppStatus nppiWarpAffineQuad_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 32-bit signed integer quad-based affine warp.*



- **NppStatus** **nppiWarpAffineQuad\_32s\_C4R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit signed integer quad-based affine warp.*

- **NppStatus** **nppiWarpAffineQuad\_32s\_AC4R** (const **Npp32s** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit signed integer quad-based affine warp, ignoring alpha channel.*

- **NppStatus** **nppiWarpAffineQuad\_32s\_P3R** (const **Npp32s** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** \*pDst[3], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 32-bit signed integer quad-based affine warp.*

- **NppStatus** **nppiWarpAffineQuad\_32s\_P4R** (const **Npp32s** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32s** \*pDst[4], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 32-bit signed integer quad-based affine warp.*

- **NppStatus** **nppiWarpAffineQuad\_32f\_C1R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 32-bit floating-point quad-based affine warp.*

- **NppStatus** **nppiWarpAffineQuad\_32f\_C3R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 32-bit floating-point quad-based affine warp.*

- **NppStatus** **nppiWarpAffineQuad\_32f\_C4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit floating-point quad-based affine warp.*

- **NppStatus** **nppiWarpAffineQuad\_32f\_AC4R** (const **Npp32f** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit floating-point quad-based affine warp, ignoring alpha channel.*

- **NppStatus** **nppiWarpAffineQuad\_32f\_P3R** (const **Npp32f** \*pSrc[3], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** \*pDst[3], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 32-bit floating-point quad-based affine warp.*

- **NppStatus** **nppiWarpAffineQuad\_32f\_P4R** (const **Npp32f** \*pSrc[4], **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, const double aSrcQuad[4][2], **Npp32f** \*pDst[4], int nDstStep, **NppiRect** oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 32-bit floating-point quad-based affine warp.*

### 7.78.1 Detailed Description

### 7.78.2 Affine Transform Error Codes

- **NPP\_RECT\_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- **NPP\_WRONG\_INTERSECTION\_ROI\_ERROR** Indicates an error condition if oSrcROI has no intersection with the source image
- **NPP\_INTERPOLATION\_ERROR** Indicates an error condition if interpolation has an illegal value
- **NPP\_COEFF\_ERROR** Indicates an error condition if coefficient values are invalid
- **NPP\_WRONG\_INTERSECTION\_QUAD\_WARNING** Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

### 7.78.3 Function Documentation

#### 7.78.3.1 NppStatus nppiGetAffineBound (NppiRect oSrcROI, double aBound[2][2], const double aCoeffs[2][3])

Compute bounding-box of transformed image.

The method effectively computes the bounding box (axis aligned rectangle) of the transformed source ROI (see [nppiGetAffineQuad\(\)](#)).

**Parameters:**

- oSrcROI** The source ROI.
- aBound** The resulting bounding box.
- aCoeffs** The affine transform coefficients.

**Returns:**

Error codes:

- **NPP\_SIZE\_ERROR** Indicates an error condition if any image dimension has zero or negative value
- **NPP\_RECT\_ERROR** Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- **NPP\_COEFF\_ERROR** Indicates an error condition if coefficient values are invalid

#### 7.78.3.2 NppStatus nppiGetAffineQuad (NppiRect oSrcROI, double aQuad[4][2], const double aCoeffs[2][3])

Compute shape of transformed image.

This method computes the quadrilateral in the destination image that the source ROI is transformed into by the affine transformation expressed by the coefficients array (aCoeffs).

**Parameters:**

- oSrcROI** The source ROI.



*aQuad* The resulting destination quadrangle.

*aCoeffs* The affine transform coefficients.

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

#### 7.78.3.3 NppStatus nppiGetAffineTransform (NppiRect oSrcROI, const double aQuad[4][2], double aCoeffs[2][3])

Computes affine transform coefficients based on source ROI and destination quadrilateral.

The function computes the coefficients of an affine transformation that maps the given source ROI (axis aligned rectangle with integer coordinates) to a quadrilateral in the destination image.

An affine transform in 2D is fully determined by the mapping of just three vertices. This function's API allows for passing a complete quadrilateral effectively making the problem overdetermined. What this means in practice is, that for certain quadrilaterals it is not possible to find an affine transform that would map all four corners of the source ROI to the four vertices of that quadrilateral.

The function circumvents this problem by only looking at the first three vertices of the destination image quadrilateral to determine the affine transformation's coefficients. If the destination quadrilateral is indeed one that cannot be mapped using an affine transformation the function informs the user of this situation by returning a [NPP\\_AFFINE\\_QUAD\\_INCORRECT\\_WARNING](#).

#### Parameters:

*oSrcROI* The source ROI. This rectangle needs to be at least one pixel wide and high. If either width or height are less than one an [NPP\\_RECT\\_ERROR](#) is returned.

*aQuad* The destination quadrilateral.

*aCoeffs* The resulting affine transform coefficients.

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the oSrcROI and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid
- [NPP\\_AFFINE\\_QUAD\\_INCORRECT\\_WARNING](#) Indicates a warning when quad does not conform to the transform properties. Fourth vertex is ignored, internally computed coordinates are used instead

**7.78.3.4** `NppStatus nppiWarpAffine_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.5** `NppStatus nppiWarpAffine_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.6** `NppStatus nppiWarpAffine_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.7** `NppStatus nppiWarpAffine_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.8** `NppStatus nppiWarpAffine_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.9** `NppStatus nppiWarpAffine_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 16-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.10** `NppStatus nppiWarpAffine_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.11** `NppStatus nppiWarpAffine_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.12** `NppStatus nppiWarpAffine_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.13** `NppStatus nppiWarpAffine_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.14** `NppStatus nppiWarpAffine_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.15** `NppStatus nppiWarpAffine_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.16** `NppStatus nppiWarpAffine_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.17** `NppStatus nppiWarpAffine_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)



**7.78.3.18** `NppStatus nppiWarpAffine_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.19** `NppStatus nppiWarpAffine_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.20** `NppStatus nppiWarpAffine_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.21** `NppStatus nppiWarpAffine_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit signed affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.22** `NppStatus nppiWarpAffine_64f_AC4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 64-bit floating-point affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.23** `NppStatus nppiWarpAffine_64f_C1R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 64-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.24** `NppStatus nppiWarpAffine_64f_C3R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 64-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.25** `NppStatus nppiWarpAffine_64f_C4R (const Npp64f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 64-bit floating-point affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.26** `NppStatus nppiWarpAffine_64f_P3R (const Npp64f * aSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * aDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 64-bit floating-point affine warp.

**Parameters:**

*aSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.27** `NppStatus nppiWarpAffine_64f_P4R (const Npp64f * aSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp64f * aDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 64-bit floating-point affine warp.

**Parameters:**

*aSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.28** `NppStatus nppiWarpAffine_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.29** `NppStatus nppiWarpAffine_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.30** `NppStatus nppiWarpAffine_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.31** `NppStatus nppiWarpAffine_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.32** `NppStatus nppiWarpAffine_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.33** `NppStatus nppiWarpAffine_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 8-bit unsigned affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)



**7.78.3.34** `NppStatus nppiWarpAffineBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.35** `NppStatus nppiWarpAffineBack_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.36** `NppStatus nppiWarpAffineBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.37** `NppStatus nppiWarpAffineBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.38** `NppStatus nppiWarpAffineBack_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.39** `NppStatus nppiWarpAffineBack_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.40** `NppStatus nppiWarpAffineBack_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.41** `NppStatus nppiWarpAffineBack_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.42** `NppStatus nppiWarpAffineBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.43** `NppStatus nppiWarpAffineBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.44** `NppStatus nppiWarpAffineBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.45** `NppStatus nppiWarpAffineBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit floating-point backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.46** `NppStatus nppiWarpAffineBack_32s_AC4R` (`const Npp32s * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect oSrcROI`, `Npp32s * pDst`, `int nDstStep`, `NppiRect oDstROI`, `const double aCoeffs[2][3]`, `int eInterpolation`)

Four-channel 32-bit signed integer backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.47** `NppStatus nppiWarpAffineBack_32s_C1R` (`const Npp32s * pSrc`, `NppiSize oSrcSize`, `int nSrcStep`, `NppiRect oSrcROI`, `Npp32s * pDst`, `int nDstStep`, `NppiRect oDstROI`, `const double aCoeffs[2][3]`, `int eInterpolation`)

Single-channel 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.48** `NppStatus nppiWarpAffineBack_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.49** `NppStatus nppiWarpAffineBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)



**7.78.3.50** `NppStatus nppiWarpAffineBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.51** `NppStatus nppiWarpAffineBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 32-bit signed integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.52** `NppStatus nppiWarpAffineBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.53** `NppStatus nppiWarpAffineBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Single-channel 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.54** `NppStatus nppiWarpAffineBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.55** `NppStatus nppiWarpAffineBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.56** `NppStatus nppiWarpAffineBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.57** `NppStatus nppiWarpAffineBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[2][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer backwards affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Affine transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.58** `NppStatus nppiWarpAffineQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.59** `NppStatus nppiWarpAffineQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.60** `NppStatus nppiWarpAffineQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.61** `NppStatus nppiWarpAffineQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.62** `NppStatus nppiWarpAffineQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.63** `NppStatus nppiWarpAffineQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 16-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.64** `NppStatus nppiWarpAffineQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.65** `NppStatus nppiWarpAffineQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)



**7.78.3.66** `NppStatus nppiWarpAffineQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.67** `NppStatus nppiWarpAffineQuad_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.68** `NppStatus nppiWarpAffineQuad_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.69** `NppStatus nppiWarpAffineQuad_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.70** `NppStatus nppiWarpAffineQuad_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.71** `NppStatus nppiWarpAffineQuad_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.72** `NppStatus nppiWarpAffineQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.73** `NppStatus nppiWarpAffineQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.74** `NppStatus nppiWarpAffineQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.75** `NppStatus nppiWarpAffineQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit signed integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.76** `NppStatus nppiWarpAffineQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based affine warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.77** `NppStatus nppiWarpAffineQuad_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.78** `NppStatus nppiWarpAffineQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.79** `NppStatus nppiWarpAffineQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.80** `NppStatus nppiWarpAffineQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)

**7.78.3.81** `NppStatus nppiWarpAffineQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 8-bit unsigned integer quad-based affine warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Affine Transform Error Codes](#)



## 7.79 Perspective Transform

### Utility Functions

- **NppStatus** **nppiGetPerspectiveTransform** (**NppiRect** oSrcROI, const double quad[4][2], double aCoeffs[3][3])

*Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.*

- **NppStatus** **nppiGetPerspectiveQuad** (**NppiRect** oSrcROI, double quad[4][2], const double aCoeffs[3][3])

*Calculates perspective transform projection of given source rectangular ROI.*

- **NppStatus** **nppiGetPerspectiveBound** (**NppiRect** oSrcROI, double bound[2][2], const double aCoeffs[3][3])

*Calculates bounding box of the perspective transform projection of the given source rectangular ROI.*

### Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective transform is given as a  $3 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates are computed as follows:

$$x' = \frac{c_{00} * x + c_{01} * y + c_{02}}{c_{20} * x + c_{21} * y + c_{22}} \quad y' = \frac{c_{10} * x + c_{11} * y + c_{12}}{c_{20} * x + c_{21} * y + c_{22}}$$

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}$$

- **NppStatus** **nppiWarpPerspective\_8u\_C1R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 8-bit unsigned integer perspective warp.*

- **NppStatus** **nppiWarpPerspective\_8u\_C3R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 8-bit unsigned integer perspective warp.*

- **NppStatus** **nppiWarpPerspective\_8u\_C4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 8-bit unsigned integer perspective warp.*

- **NppStatus** **nppiWarpPerspective\_8u\_AC4R** (const **Npp8u** \*pSrc, **NppiSize** oSrcSize, int nSrcStep, **NppiRect** oSrcROI, **Npp8u** \*pDst, int nDstStep, **NppiRect** oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 8-bit unsigned integer perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspective_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel planar 8-bit unsigned integer perspective warp.*

- `NppStatus nppiWarpPerspective_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 8-bit unsigned integer perspective warp.*

- `NppStatus nppiWarpPerspective_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 16-bit unsigned integer perspective warp.*

- `NppStatus nppiWarpPerspective_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 16-bit unsigned integer perspective warp.*

- `NppStatus nppiWarpPerspective_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 16-bit unsigned integer perspective warp.*

- `NppStatus nppiWarpPerspective_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 16-bit unsigned integer perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspective_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel planar 16-bit unsigned integer perspective warp.*

- `NppStatus nppiWarpPerspective_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 16-bit unsigned integer perspective warp.*

- `NppStatus nppiWarpPerspective_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 32-bit signed integer perspective warp.*

- `NppStatus nppiWarpPerspective_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 32-bit signed integer perspective warp.*

- `NppStatus nppiWarpPerspective_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit signed integer perspective warp.*

- `NppStatus nppiWarpPerspective_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit signed integer perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspective_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel planar 32-bit signed integer perspective warp.*

- `NppStatus nppiWarpPerspective_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 32-bit signed integer perspective warp.*

- `NppStatus nppiWarpPerspective_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 32-bit floating-point perspective warp.*

- `NppStatus nppiWarpPerspective_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 32-bit floating-point perspective warp.*

- `NppStatus nppiWarpPerspective_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit floating-point perspective warp.*

- `NppStatus nppiWarpPerspective_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit floating-point perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspective_32f_P3R` (const `Npp32f` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel planar 32-bit floating-point perspective warp.*

- `NppStatus nppiWarpPerspective_32f_P4R` (const `Npp32f` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 32-bit floating-point perspective warp.*

## Backwards Perspective Transform

Transforms (warps) an image based on a perspective transform.

The perspective transform is given as a  $3 \times 3$  matrix  $C$ . A pixel location  $(x, y)$  in the source image is mapped to the location  $(x', y')$  in the destination image. The destination image coordinates fulfill the following properties:

$$x = \frac{c_{00} * x' + c_{01} * y' + c_{02}}{c_{20} * x' + c_{21} * y' + c_{22}} \quad y = \frac{c_{10} * x' + c_{11} * y' + c_{12}}{c_{20} * x' + c_{21} * y' + c_{22}}$$

$$C = \begin{bmatrix} c_{00} & c_{01} & c_{02} \\ c_{10} & c_{11} & c_{12} \\ c_{20} & c_{21} & c_{22} \end{bmatrix}$$

In other words, given matrix  $C$  the source image's shape is transformed to the destination image using the inverse matrix  $C^{-1}$ :

$$M = C^{-1} = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \\ m_{20} & m_{21} & m_{22} \end{bmatrix} \quad x' = \frac{c_{00} * x + c_{01} * y + c_{02}}{c_{20} * x + c_{21} * y + c_{22}} \quad y' = \frac{c_{10} * x + c_{11} * y + c_{12}}{c_{20} * x + c_{21} * y + c_{22}}$$

- `NppStatus nppiWarpPerspectiveBack_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 8-bit unsigned integer backwards perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveBack_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel planar 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp8u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 8-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 16-bit unsigned integer backwards perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveBack_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp16u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 16-bit unsigned integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit signed integer backwards perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveBack_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel planar 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32s` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 32-bit signed integer backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Single-channel 32-bit floating-point backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel 32-bit floating-point backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit floating-point backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel 32-bit floating-point backwards perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveBack_32f_P3R` (const `Npp32f` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Three-channel planar 32-bit floating-point backwards perspective warp.*

- `NppStatus nppiWarpPerspectiveBack_32f_P4R` (const `Npp32f` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, `Npp32f` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aCoeffs[3][3], int eInterpolation)

*Four-channel planar 32-bit floating-point backwards perspective warp.*

## Quad-Based Perspective Transform

Transforms (warps) an image based on an perspective transform.

The perspective transform is computed such that it maps a quadrilateral in source image space to a quadrilateral in destination image space.

- `NppStatus nppiWarpPerspectiveQuad_8u_C1R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_8u_C3R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_8u_C4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_8u_AC4R` (const `Npp8u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 8-bit unsigned integer quad-based perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveQuad_8u_P3R` (const `Npp8u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_8u_P4R` (const `Npp8u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp8u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 8-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_C1R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_C3R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_C4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_AC4R` (const `Npp16u` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 16-bit unsigned integer quad-based perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveQuad_16u_P3R` (const `Npp16u` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 16-bit unsigned integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_16u_P4R` (const `Npp16u` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp16u` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 16-bit unsigned integer quad-based perspective warp.*



- `NppStatus nppiWarpPerspectiveQuad_32s_C1R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 32-bit signed integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32s_C3R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 32-bit signed integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32s_C4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit signed integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32s_AC4R` (const `Npp32s` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit signed integer quad-based perspective warp, ignoring alpha channel.*

- `NppStatus nppiWarpPerspectiveQuad_32s_P3R` (const `Npp32s` \*pSrc[3], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst[3], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel planar 32-bit signed integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32s_P4R` (const `Npp32s` \*pSrc[4], `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32s` \*pDst[4], int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel planar 32-bit signed integer quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32f_C1R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Single-channel 32-bit floating-point quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32f_C3R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Three-channel 32-bit floating-point quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32f_C4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit floating-point quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32f_AC4R` (const `Npp32f` \*pSrc, `NppiSize` oSrcSize, int nSrcStep, `NppiRect` oSrcROI, const double aSrcQuad[4][2], `Npp32f` \*pDst, int nDstStep, `NppiRect` oDstROI, const double aDstQuad[4][2], int eInterpolation)

*Four-channel 32-bit floating-point quad-based perspective warp, ignoring alpha channel.*



- `NppStatus nppiWarpPerspectiveQuad_32f_P3R` (const `Npp32f *pSrc[3]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[3]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Three-channel planar 32-bit floating-point quad-based perspective warp.*

- `NppStatus nppiWarpPerspectiveQuad_32f_P4R` (const `Npp32f *pSrc[4]`, `NppiSize` `oSrcSize`, int `nSrcStep`, `NppiRect` `oSrcROI`, const double `aSrcQuad[4][2]`, `Npp32f *pDst[4]`, int `nDstStep`, `NppiRect` `oDstROI`, const double `aDstQuad[4][2]`, int `eInterpolation`)

*Four-channel planar 32-bit floating-point quad-based perspective warp.*

### 7.79.1 Detailed Description

### 7.79.2 Perspective Transform Error Codes

- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `oSrcROI` and source image is less than or equal to 1
- `NPP_WRONG_INTERSECTION_ROI_ERROR` Indicates an error condition if `oSrcROI` has no intersection with the source image
- `NPP_INTERPOLATION_ERROR` Indicates an error condition if interpolation has an illegal value
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid
- `NPP_WRONG_INTERSECTION_QUAD_WARNING` Indicates a warning that no operation is performed if the transformed source ROI has no intersection with the destination ROI

### 7.79.3 Function Documentation

#### 7.79.3.1 `NppStatus nppiGetPerspectiveBound (NppiRect oSrcROI, double bound[2][2], const double aCoeffs[3][3])`

Calculates bounding box of the perspective transform projection of the given source rectangular ROI.

#### Parameters:

*oSrcROI* Source ROI

*bound* Bounding box of the transformed source ROI

*aCoeffs* Perspective transform coefficients

#### Returns:

Error codes:

- `NPP_SIZE_ERROR` Indicates an error condition if any image dimension has zero or negative value
- `NPP_RECT_ERROR` Indicates an error condition if width or height of the intersection of the `oSrcROI` and source image is less than or equal to 1
- `NPP_COEFF_ERROR` Indicates an error condition if coefficient values are invalid

### 7.79.3.2 NppStatus nppiGetPerspectiveQuad (NppiRect *oSrcROI*, double *quad*[4][2], const double *aCoeffs*[3][3])

Calculates perspective transform projection of given source rectangular ROI.

#### Parameters:

*oSrcROI* Source ROI  
*quad* Destination quadrangle  
*aCoeffs* Perspective transform coefficients

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *oSrcROI* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

### 7.79.3.3 NppStatus nppiGetPerspectiveTransform (NppiRect *oSrcROI*, const double *quad*[4][2], double *aCoeffs*[3][3])

Calculates perspective transform coefficients given source rectangular ROI and its destination quadrangle projection.

#### Parameters:

*oSrcROI* Source ROI  
*quad* Destination quadrangle  
*aCoeffs* Perspective transform coefficients

#### Returns:

Error codes:

- [NPP\\_SIZE\\_ERROR](#) Indicates an error condition if any image dimension has zero or negative value
- [NPP\\_RECT\\_ERROR](#) Indicates an error condition if width or height of the intersection of the *oSrcROI* and source image is less than or equal to 1
- [NPP\\_COEFF\\_ERROR](#) Indicates an error condition if coefficient values are invalid

### 7.79.3.4 NppStatus nppiWarpPerspective\_16u\_AC4R (const Npp16u \* *pSrc*, NppiSize *oSrcSize*, int *nSrcStep*, NppiRect *oSrcROI*, Npp16u \* *pDst*, int *nDstStep*, NppiRect *oDstROI*, const double *aCoeffs*[3][3], int *eInterpolation*)

Four-channel 16-bit unsigned integer perspective warp, ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcROI* Source ROI  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.5** `NppStatus nppiWarpPerspective_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*oSrcSize* Size of source image in pixels  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcROI* Source ROI  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oDstROI* Destination ROI  
*aCoeffs* Perspective transform coefficients  
*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.6** `NppStatus nppiWarpPerspective_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.7** `NppStatus nppiWarpPerspective_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.8** `NppStatus nppiWarpPerspective_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 16-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.79.3.9** `NppStatus nppiWarpPerspective_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer perspective warp.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.79.3.10** `NppStatus nppiWarpPerspective_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point perspective warp, ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.79.3.11** `NppStatus nppiWarpPerspective_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.79.3.12** `NppStatus nppiWarpPerspective_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.79.3.13** `NppStatus nppiWarpPerspective_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point perspective warp.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, Perspective Transform Error Codes

**7.79.3.14** `NppStatus nppiWarpPerspective_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit floating-point perspective warp.

#### Parameters:

*pSrc* Source-Image Pointer.

*oSrcSize* Size of source image in pixels

*nSrcStep* Source-Image Line Step.

*oSrcROI* Source ROI

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.15** `NppStatus nppiWarpPerspective_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit floating-point perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.16** `NppStatus nppiWarpPerspective_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI



*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.17** `NppStatus nppiWarpPerspective_32s_C1R (const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.18** `NppStatus nppiWarpPerspective_32s_C3R (const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.19** `NppStatus nppiWarpPerspective_32s_C4R (const Npp32s *pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.20** `NppStatus nppiWarpPerspective_32s_P3R (const Npp32s *pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s *pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.21** `NppStatus nppiWarpPerspective_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit signed integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.22** `NppStatus nppiWarpPerspective_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.23** `NppStatus nppiWarpPerspective_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.24** `NppStatus nppiWarpPerspective_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.25** `NppStatus nppiWarpPerspective_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.26** `NppStatus nppiWarpPerspective_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.27** `NppStatus nppiWarpPerspective_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.28** `NppStatus nppiWarpPerspectiveBack_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.29** `NppStatus nppiWarpPerspectiveBack_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.30** `NppStatus nppiWarpPerspectiveBack_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.31** `NppStatus nppiWarpPerspectiveBack_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.32** `NppStatus nppiWarpPerspectiveBack_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)



**7.79.3.33** `NppStatus nppiWarpPerspectiveBack_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 16-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.34** `NppStatus nppiWarpPerspectiveBack_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.35** `NppStatus nppiWarpPerspectiveBack_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.36** `NppStatus nppiWarpPerspectiveBack_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.37** `NppStatus nppiWarpPerspectiveBack_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.38** `NppStatus nppiWarpPerspectiveBack_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.39** `NppStatus nppiWarpPerspectiveBack_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit floating-point backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.40** `NppStatus nppiWarpPerspectiveBack_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.41** `NppStatus nppiWarpPerspectiveBack_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.42** `NppStatus nppiWarpPerspectiveBack_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.43** `NppStatus nppiWarpPerspectiveBack_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.44** `NppStatus nppiWarpPerspectiveBack_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.45** `NppStatus nppiWarpPerspectiveBack_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 32-bit signed integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.46** `NppStatus nppiWarpPerspectiveBack_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.47** `NppStatus nppiWarpPerspectiveBack_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Single-channel 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.48** `NppStatus nppiWarpPerspectiveBack_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)



**7.79.3.49** `NppStatus nppiWarpPerspectiveBack_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.50** `NppStatus nppiWarpPerspectiveBack_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Three-channel planar 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.51** `NppStatus nppiWarpPerspectiveBack_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aCoeffs[3][3], int eInterpolation)`

Four-channel planar 8-bit unsigned integer backwards perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aCoeffs* Perspective transform coefficients

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.52** `NppStatus nppiWarpPerspectiveQuad_16u_AC4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.53** `NppStatus nppiWarpPerspectiveQuad_16u_C1R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.54** `NppStatus nppiWarpPerspectiveQuad_16u_C3R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.55** `NppStatus nppiWarpPerspectiveQuad_16u_C4R (const Npp16u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.56** `NppStatus nppiWarpPerspectiveQuad_16u_P3R (const Npp16u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.57** `NppStatus nppiWarpPerspectiveQuad_16u_P4R (const Npp16u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp16u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 16-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.58** `NppStatus nppiWarpPerspectiveQuad_32f_AC4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.59** `NppStatus nppiWarpPerspectiveQuad_32f_C1R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.60** `NppStatus nppiWarpPerspectiveQuad_32f_C3R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.61** `NppStatus nppiWarpPerspectiveQuad_32f_C4R (const Npp32f * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.62** `NppStatus nppiWarpPerspectiveQuad_32f_P3R (const Npp32f * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.63** `NppStatus nppiWarpPerspectiveQuad_32f_P4R (const Npp32f * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32f * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit floating-point quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.64** `NppStatus nppiWarpPerspectiveQuad_32s_AC4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)



**7.79.3.65** `NppStatus nppiWarpPerspectiveQuad_32s_C1R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.66** `NppStatus nppiWarpPerspectiveQuad_32s_C3R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.67** `NppStatus nppiWarpPerspectiveQuad_32s_C4R (const Npp32s * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.68** `NppStatus nppiWarpPerspectiveQuad_32s_P3R (const Npp32s * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.69** `NppStatus nppiWarpPerspectiveQuad_32s_P4R (const Npp32s * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp32s * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 32-bit signed integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.70** `NppStatus nppiWarpPerspectiveQuad_8u_AC4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based perspective warp, ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.71** `NppStatus nppiWarpPerspectiveQuad_8u_C1R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Single-channel 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.72** `NppStatus nppiWarpPerspectiveQuad_8u_C3R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.73** `NppStatus nppiWarpPerspectiveQuad_8u_C4R (const Npp8u * pSrc, NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst, int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.74** `NppStatus nppiWarpPerspectiveQuad_8u_P3R (const Npp8u * pSrc[3], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[3], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Three-channel planar 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

**7.79.3.75** `NppStatus nppiWarpPerspectiveQuad_8u_P4R (const Npp8u * pSrc[4], NppiSize oSrcSize, int nSrcStep, NppiRect oSrcROI, const double aSrcQuad[4][2], Npp8u * pDst[4], int nDstStep, NppiRect oDstROI, const double aDstQuad[4][2], int eInterpolation)`

Four-channel planar 8-bit unsigned integer quad-based perspective warp.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*oSrcSize* Size of source image in pixels

*nSrcStep* [Source-Image Line Step](#).

*oSrcROI* Source ROI

*aSrcQuad* Source quad.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oDstROI* Destination ROI

*aDstQuad* Destination quad.

*eInterpolation* Interpolation mode: can be NPPI\_INTER\_NN, NPPI\_INTER\_LINEAR or NPPI\_INTER\_CUBIC

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), [Perspective Transform Error Codes](#)

## 7.80 Linear Transforms

Linear image transformations.

### Modules

- [Fourier Transforms](#)

### 7.80.1 Detailed Description

Linear image transformations.

## 7.81 Fourier Transforms

### Functions

- **NppStatus nppiMagnitude\_32fc32f\_C1R** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit floating point complex to 32-bit floating point magnitude.*
- **NppStatus nppiMagnitudeSqr\_32fc32f\_C1R** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*32-bit floating point complex to 32-bit floating point squared magnitude.*

### 7.81.1 Function Documentation

#### 7.81.1.1 **NppStatus nppiMagnitude\_32fc32f\_C1R** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

32-bit floating point complex to 32-bit floating point magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the magnitude of the complex values.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.81.1.2 **NppStatus nppiMagnitudeSqr\_32fc32f\_C1R** (const **Npp32fc** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)

32-bit floating point complex to 32-bit floating point squared magnitude.

Converts complex-number pixel image to single channel image computing the result pixels as the squared magnitude of the complex values.

The squared magnitude is an intermediate result of magnitude computation and can thus be computed faster than actual magnitude. If magnitudes are required for sorting/comparing only, using this function instead of `nppiMagnitude_32fc32f_C1R` can be a worthwhile performance optimization.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.



*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.82 Morphological Operations

Morphological image operations.

### Modules

- [Dilation](#)

*Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.*

- [Erode](#)

*Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.*

- [Dilate3x3](#)

*Dilation using a 3x3 mask with the anchor at its center pixel.*

- [Erode3x3](#)

*Erosion using a 3x3 mask with the anchor at its center pixel.*

### 7.82.1 Detailed Description

Morphological image operations.

Morphological operations are classified as [Neighborhood Operations](#). It is the user's responsibility to avoid [Sampling Beyond Image Boundaries](#).

## 7.83 Dilation

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

### Functions

- **NppStatus nppiDilate\_8u\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single-channel 8-bit unsigned integer dilation.*
- **NppStatus nppiDilate\_8u\_C3R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three-channel 8-bit unsigned integer dilation.*
- **NppStatus nppiDilate\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 8-bit unsigned integer dilation.*
- **NppStatus nppiDilate\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 8-bit unsigned integer dilation, ignoring alpha-channel.*
- **NppStatus nppiDilate\_16u\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single-channel 16-bit unsigned integer dilation.*
- **NppStatus nppiDilate\_16u\_C3R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three-channel 16-bit unsigned integer dilation.*
- **NppStatus nppiDilate\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 16-bit unsigned integer dilation.*
- **NppStatus nppiDilate\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 16-bit unsigned integer dilation, ignoring alpha-channel.*
- **NppStatus nppiDilate\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single-channel 32-bit floating-point dilation.*
- **NppStatus nppiDilate\_32f\_C3R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three-channel 32-bit floating-point dilation.*
- **NppStatus nppiDilate\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 32-bit floating-point dilation.*

- **NppStatus nppiDilate\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

*Four-channel 32-bit floating-point dilation, ignoring alpha-channel.*

### 7.83.1 Detailed Description

Dilation computes the output pixel as the maximum pixel value of the pixels under the mask.

Pixels whose corresponding mask values are zero do not participate in the maximum search.

### 7.83.2 Function Documentation

- 7.83.2.1 NppStatus nppiDilate\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Four-channel 16-bit unsigned integer dilation, ignoring alpha-channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.83.2.2 NppStatus nppiDilate\_16u\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Single-channel 16-bit unsigned integer dilation.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.3** `NppStatus nppiDilate_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 16-bit unsigned integer dilation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.4** `NppStatus nppiDilate_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 16-bit unsigned integer dilation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.5** `NppStatus nppiDilate_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point dilation, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.6** `NppStatus nppiDilate_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 32-bit floating-point dilation.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.7** `NppStatus nppiDilate_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 32-bit floating-point dilation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Pointer to the start address of the mask array  
*oMaskSize* Width and Height mask array.  
*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.8** `NppStatus nppiDilate_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point dilation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Pointer to the start address of the mask array  
*oMaskSize* Width and Height mask array.  
*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.9** `NppStatus nppiDilate_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer dilation, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.10** `NppStatus nppiDilate_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 8-bit unsigned integer dilation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.83.2.11** `NppStatus nppiDilate_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 8-bit unsigned integer dilation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.83.2.12** `NppStatus nppiDilate_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer dilation.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.84 Erode

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

### Functions

- **NppStatus nppiErode\_8u\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single-channel 8-bit unsigned integer erosion.*
- **NppStatus nppiErode\_8u\_C3R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three-channel 8-bit unsigned integer erosion.*
- **NppStatus nppiErode\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 8-bit unsigned integer erosion.*
- **NppStatus nppiErode\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 8-bit unsigned integer erosion, ignoring alpha-channel.*
- **NppStatus nppiErode\_16u\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single-channel 16-bit unsigned integer erosion.*
- **NppStatus nppiErode\_16u\_C3R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three-channel 16-bit unsigned integer erosion.*
- **NppStatus nppiErode\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 16-bit unsigned integer erosion.*
- **NppStatus nppiErode\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 16-bit unsigned integer erosion, ignoring alpha-channel.*
- **NppStatus nppiErode\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Single-channel 32-bit floating-point erosion.*
- **NppStatus nppiErode\_32f\_C3R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Three-channel 32-bit floating-point erosion.*
- **NppStatus nppiErode\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)  
*Four-channel 32-bit floating-point erosion.*

- **NppStatus nppiErode\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

*Four-channel 32-bit floating-point erosion, ignoring alpha-channel.*

### 7.84.1 Detailed Description

Erosion computes the output pixel as the minimum pixel value of the pixels under the mask.

Pixels whose corresponding mask values are zero do not participate in the maximum search.

### 7.84.2 Function Documentation

- 7.84.2.1 NppStatus nppiErode\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Four-channel 16-bit unsigned integer erosion, ignoring alpha-channel.

#### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pDst** Destination-Image Pointer.

**nDstStep** Destination-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

**pMask** Pointer to the start address of the mask array

**oMaskSize** Width and Height mask array.

**oAnchor** X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.84.2.2 NppStatus nppiErode\_16u\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI, const **Npp8u** \*pMask, **NppiSize** oMaskSize, **NppiPoint** oAnchor)

Single-channel 16-bit unsigned integer erosion.

#### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pDst** Destination-Image Pointer.

**nDstStep** Destination-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

**pMask** Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.3** `NppStatus nppiErode_16u_C3R (const Npp16u * pSrc, Npp32s nSrcStep, Npp16u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 16-bit unsigned integer erosion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.4** `NppStatus nppiErode_16u_C4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 16-bit unsigned integer erosion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.5** `NppStatus nppiErode_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point erosion, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.6** `NppStatus nppiErode_32f_C1R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 32-bit floating-point erosion.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.7** `NppStatus nppiErode_32f_C3R (const Npp32f * pSrc, Npp32s nSrcStep, Npp32f * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 32-bit floating-point erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Pointer to the start address of the mask array  
*oMaskSize* Width and Height mask array.  
*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.8** `NppStatus nppiErode_32f_C4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 32-bit floating-point erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pMask* Pointer to the start address of the mask array  
*oMaskSize* Width and Height mask array.  
*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.9** `NppStatus nppiErode_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer erosion, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.10** `NppStatus nppiErode_8u_C1R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Single-channel 8-bit unsigned integer erosion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.11** `NppStatus nppiErode_8u_C3R (const Npp8u * pSrc, Npp32s nSrcStep, Npp8u * pDst, Npp32s nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Three-channel 8-bit unsigned integer erosion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.84.2.12** `NppStatus nppiErode_8u_C4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u * pMask, NppiSize oMaskSize, NppiPoint oAnchor)`

Four-channel 8-bit unsigned integer erosion.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMask* Pointer to the start address of the mask array

*oMaskSize* Width and Height mask array.

*oAnchor* X and Y offsets of the mask origin frame of reference w.r.t the source pixel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



## 7.85 Dilate3x3

Dilation using a 3x3 mask with the anchor at its center pixel.

### Functions

- **NppStatus** **nppiDilate3x3\_8u\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 8-bit unsigned integer 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_8u\_C3R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Three-channel 8-bit unsigned integer 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit unsigned integer 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit unsigned integer 3x3 dilation, ignoring alpha-channel.*
- **NppStatus** **nppiDilate3x3\_16u\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 16-bit unsigned integer 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_16u\_C3R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit unsigned integer 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit unsigned integer 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit unsigned integer 3x3 dilation, ignoring alpha-channel.*
- **NppStatus** **nppiDilate3x3\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 32-bit floating-point 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_32f\_C3R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Three-channel 32-bit floating-point 3x3 dilation.*
- **NppStatus** **nppiDilate3x3\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit floating-point 3x3 dilation.*

- **NppStatus nppiDilate3x3\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit floating-point 3x3 dilation, ignoring alpha-channel.*
- **NppStatus nppiDilate3x3\_64f\_C1R** (const **Npp64f** \*pSrc, **Npp32s** nSrcStep, **Npp64f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 64-bit floating-point 3x3 dilation.*

### 7.85.1 Detailed Description

Dilation using a 3x3 mask with the anchor at its center pixel.

### 7.85.2 Function Documentation

#### 7.85.2.1 NppStatus nppiDilate3x3\_16u\_AC4R (const Npp16u \*pSrc, int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)

Four-channel 16-bit unsigned integer 3x3 dilation, ignoring alpha-channel.

##### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.85.2.2 NppStatus nppiDilate3x3\_16u\_C1R (const Npp16u \*pSrc, Npp32s nSrcStep, Npp16u \*pDst, Npp32s nDstStep, NppiSize oSizeROI)

Single-channel 16-bit unsigned integer 3x3 dilation.

##### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.85.2.3 NppStatus nppiDilate3x3\_16u\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 16-bit unsigned integer 3x3 dilation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.85.2.4 NppStatus nppiDilate3x3\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 16-bit unsigned integer 3x3 dilation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.85.2.5 NppStatus nppiDilate3x3\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit floating-point 3x3 dilation, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

### 7.85.2.6 NppStatus nppiDilate3x3\_32f\_C1R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 32-bit floating-point 3x3 dilation.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.85.2.7 NppStatus nppiDilate3x3\_32f\_C3R (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit floating-point 3x3 dilation.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.85.2.8 NppStatus nppiDilate3x3\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point 3x3 dilation.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.85.2.9 NppStatus nppiDilate3x3\_64f\_C1R (const Npp64f \* *pSrc*, Npp32s *nSrcStep*, Npp64f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 64-bit floating-point 3x3 dilation.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.85.2.10 NppStatus nppiDilate3x3\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 8-bit unsigned integer 3x3 dilation, ignoring alpha-channel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.85.2.11 NppStatus nppiDilate3x3\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 8-bit unsigned integer 3x3 dilation.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.85.2.12 NppStatus nppiDilate3x3\_8u\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 8-bit unsigned integer 3x3 dilation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.85.2.13 NppStatus nppiDilate3x3\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit unsigned integer 3x3 dilation.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.86 Erode3x3

Erosion using a 3x3 mask with the anchor at its center pixel.

### Functions

- **NppStatus nppiErode3x3\_8u\_C1R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 8-bit unsigned integer 3x3 erosion.*
- **NppStatus nppiErode3x3\_8u\_C3R** (const **Npp8u** \*pSrc, **Npp32s** nSrcStep, **Npp8u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Three-channel 8-bit unsigned integer 3x3 erosion.*
- **NppStatus nppiErode3x3\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit unsigned integer 3x3 erosion.*
- **NppStatus nppiErode3x3\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 8-bit unsigned integer 3x3 erosion, ignoring alpha-channel.*
- **NppStatus nppiErode3x3\_16u\_C1R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 16-bit unsigned integer 3x3 erosion.*
- **NppStatus nppiErode3x3\_16u\_C3R** (const **Npp16u** \*pSrc, **Npp32s** nSrcStep, **Npp16u** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Three-channel 16-bit unsigned integer 3x3 erosion.*
- **NppStatus nppiErode3x3\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit unsigned integer 3x3 erosion.*
- **NppStatus nppiErode3x3\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 16-bit unsigned integer 3x3 erosion, ignoring alpha-channel.*
- **NppStatus nppiErode3x3\_32f\_C1R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 32-bit floating-point 3x3 erosion.*
- **NppStatus nppiErode3x3\_32f\_C3R** (const **Npp32f** \*pSrc, **Npp32s** nSrcStep, **Npp32f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Three-channel 32-bit floating-point 3x3 erosion.*
- **NppStatus nppiErode3x3\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit floating-point 3x3 erosion.*

- **NppStatus nppiErode3x3\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI)  
*Four-channel 32-bit floating-point 3x3 erosion, ignoring alpha-channel.*
- **NppStatus nppiErode3x3\_64f\_C1R** (const **Npp64f** \*pSrc, **Npp32s** nSrcStep, **Npp64f** \*pDst, **Npp32s** nDstStep, **NppiSize** oSizeROI)  
*Single-channel 64-bit floating-point 3x3 erosion.*

### 7.86.1 Detailed Description

Erosion using a 3x3 mask with the anchor at its center pixel.

### 7.86.2 Function Documentation

#### 7.86.2.1 NppStatus nppiErode3x3\_16u\_AC4R (const Npp16u \*pSrc, int nSrcStep, Npp16u \*pDst, int nDstStep, NppiSize oSizeROI)

Four-channel 16-bit unsigned integer 3x3 erosion, ignoring alpha-channel.

##### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.86.2.2 NppStatus nppiErode3x3\_16u\_C1R (const Npp16u \*pSrc, Npp32s nSrcStep, Npp16u \*pDst, Npp32s nDstStep, NppiSize oSizeROI)

Single-channel 16-bit unsigned integer 3x3 erosion.

##### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.  
**oSizeROI** Region-of-Interest (ROI).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.86.2.3 NppStatus nppiErode3x3\_16u\_C3R (const Npp16u \* *pSrc*, Npp32s *nSrcStep*, Npp16u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 16-bit unsigned integer 3x3 erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.86.2.4 NppStatus nppiErode3x3\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 16-bit unsigned integer 3x3 erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.86.2.5 NppStatus nppiErode3x3\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit floating-point 3x3 erosion, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.86.2.6 **NppStatus nppiErode3x3\_32f\_C1R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Single-channel 32-bit floating-point 3x3 erosion.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.86.2.7 **NppStatus nppiErode3x3\_32f\_C3R** (const Npp32f \* *pSrc*, Npp32s *nSrcStep*, Npp32f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)

Three-channel 32-bit floating-point 3x3 erosion.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.86.2.8 **NppStatus nppiErode3x3\_32f\_C4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)

Four-channel 32-bit floating-point 3x3 erosion.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.86.2.9 NppStatus nppiErode3x3\_64f\_C1R (const Npp64f \* *pSrc*, Npp32s *nSrcStep*, Npp64f \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single-channel 64-bit floating-point 3x3 erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.86.2.10 NppStatus nppiErode3x3\_8u\_AC4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit unsigned integer 3x3 erosion, ignoring alpha-channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.86.2.11 NppStatus nppiErode3x3\_8u\_C1R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Single-channel 8-bit unsigned integer 3x3 erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.86.2.12 NppStatus nppiErode3x3\_8u\_C3R (const Npp8u \* *pSrc*, Npp32s *nSrcStep*, Npp8u \* *pDst*, Npp32s *nDstStep*, NppiSize *oSizeROI*)**

Three-channel 8-bit unsigned integer 3x3 erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.86.2.13 NppStatus nppiErode3x3\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit unsigned integer 3x3 erosion.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.87 Statistical Operations

Primitives for computing the statistical properties of an image.

### Modules

- [Sum](#)

*Primitives for computing the sum of all the pixel values in an image.*

- [Min](#)

*Primitives for computing the minimal pixel value of an image.*

- [MinIndx](#)

*Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.*

- [Max](#)

*Primitives for computing the maximal pixel value of an image.*

- [MaxIndx](#)

*Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.*

- [MinMax](#)

*Primitives for computing both the minimal and the maximal values of an image.*

- [MinMaxIndx](#)

*Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.*

- [Mean](#)

*Primitives for computing the arithmetic mean of all the pixel values in an image.*

- [Mean\\_StdDev](#)

*Primitives for computing both the arithmetic mean and the standard deviation of an image.*

- [Image Norms](#)

*Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.*

- [DotProd](#)

*Primitives for computing the dot product of two images.*

- [CountInRange.](#)

*Primitives for computing the amount of pixels that fall into the specified intensity range.*

- [MaxEvery](#)

*Primitives for computing the maximal value of the pixel pair from two images.*

- [MinEvery](#)

*Primitives for computing the minimal value of the pixel pair from two images.*

- [Integral](#)

*Primitives for computing the integral image of a given image.*

- [SqrIntegral](#)

*Primitives for computing both the integral and the squared integral images of a given image.*

- [RectStdDev](#)

*Primitives for computing the standard deviation of the integral images.*

- [HistogramEven](#)

*Primitives for computing the histogram of an image with evenly distributed bins.*

- [HistogramRange](#)

*Primitives for computing the histogram of an image within specified ranges.*

- [Image Proximity](#)

*Primitives for computing the proximity measure between a source image and a template image.*

- [Image Quality Index](#)

*Primitives for computing the image quality index of two images.*

- [MaximumError](#)

*Primitives for computing the maximum error between two images.*

- [AverageError](#)

*Primitives for computing the average error between two images.*

- [MaximumRelativeError](#)

*Primitives for computing the maximum relative error between two images.*

- [AverageRelativeError](#)

*Primitives for computing the average relative error between two images.*

## NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff\_Inf primitives.

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize\\_8u\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))

*Buffer size for [nppiMaximumError\\_8u\\_C1R](#).*

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize\\_8s\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))

*Buffer size for [nppiMaximumError\\_8s\\_C1R](#).*

- [NppStatus](#) [nppiMaximumErrorGetBufferHostSize\\_16u\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))

*Buffer size for `nppiMaximumError_16u_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16s_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_16s_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16sc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_16sc_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32u_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_32u_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32s_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_32s_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32sc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_32sc_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_32f_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_32fc_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_32fc_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_64f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_64f_C1R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_8u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_8u_C2R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_8s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_8s_C2R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16u_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_16u_C2R`.*

- `NppStatus` `nppiMaximumErrorGetBufferHostSize_16s_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiMaximumError_16s_C2R`.*

- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_16sc\\_C2R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_16sc\\_C2R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_32u\\_C2R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_32u\\_C2R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_32s\\_C2R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_32s\\_C2R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_32sc\\_C2R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_32sc\\_C2R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_32f\\_C2R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_32f\\_C2R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_32fc\\_C2R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_32fc\\_C2R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_64f\\_C2R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_64f\\_C2R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_8u\\_C3R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_8u\\_C3R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_8s\\_C3R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_8s\\_C3R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_16u\\_C3R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_16u\\_C3R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_16s\\_C3R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_16s\\_C3R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_16sc\\_C3R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_16sc\\_C3R](#).*
- [NppStatus](#)    [nppiMaximumErrorGetBufferHostSize\\_32u\\_C3R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#)  
\*hpBufferSize)  
    *Buffer size for [nppiMaximumError\\_32u\\_C3R](#).*



- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_32s\_C3R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_32s\_C3R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_32sc\_C3R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_32sc\_C3R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_32f\_C3R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_32f\_C3R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_32fc\_C3R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_32fc\_C3R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_64f\_C3R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_64f\_C3R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_8u\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_8u\_C4R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_8s\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_8s\_C4R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_16u\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_16u\_C4R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_16s\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_16s\_C4R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_16sc\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_16sc\_C4R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_32u\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_32u\_C4R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_32s\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)  
                   *Buffer size for nppiMaximumError\_32s\_C4R.*
- **NppStatus**    **nppiMaximumErrorGetBufferHostSize\_32sc\_C4R**    (**NppiSize**    **oSizeROI**,    **int**  
                   \*hpBufferSize)

*Buffer size for `nppiMaximumError_32sc_C4R`.*

- `NppStatus nppiMaximumErrorGetBufferHostSize_32f_C4R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumError_32f_C4R`.*

- `NppStatus nppiMaximumErrorGetBufferHostSize_32fc_C4R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumError_32fc_C4R`.*

- `NppStatus nppiMaximumErrorGetBufferHostSize_64f_C4R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumError_64f_C4R`.*

## NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff\_Inf primitives.

- `NppStatus nppiAverageErrorGetBufferHostSize_8u_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_8u_C1R`.*

- `NppStatus nppiAverageErrorGetBufferHostSize_8s_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_8s_C1R`.*

- `NppStatus nppiAverageErrorGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_16u_C1R`.*

- `NppStatus nppiAverageErrorGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_16s_C1R`.*

- `NppStatus nppiAverageErrorGetBufferHostSize_16sc_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_16sc_C1R`.*

- `NppStatus nppiAverageErrorGetBufferHostSize_32u_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_32u_C1R`.*

- `NppStatus nppiAverageErrorGetBufferHostSize_32s_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_32s_C1R`.*

- `NppStatus nppiAverageErrorGetBufferHostSize_32sc_C1R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiAverageError_32sc_C1R`.*

- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C1R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_32f_C1R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C1R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_32fc_C1R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C1R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_64f_C1R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_8u_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_8s_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_16u_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_16s_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_16sc_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_32u_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_32s_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_32sc_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_32f_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C2R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
*Buffer size for `nppiAverageError_32fc_C2R`.*

- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C2R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_64f_C2R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_8u_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_8s_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_16u_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_16s_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_16sc_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_32u_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_32s_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_32sc_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_32f_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_32fc_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C3R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_64f_C3R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8u_C4R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_8u_C4R`.*
- **NppStatus** `nppiAverageErrorGetBufferHostSize_8s_C4R` (**NppiSize** `oSizeROI`, **int** `*hpBufferSize`)  
*Buffer size for `nppiAverageError_8s_C4R`.*

- **NppStatus** `nppiAverageErrorGetBufferHostSize_16u_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_16u\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16s_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_16s\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_16sc_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_16sc\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32u_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_32u\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32s_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_32s\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32sc_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_32sc\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32f_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_32f\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_32fc_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_32fc\_C4R*.
- **NppStatus** `nppiAverageErrorGetBufferHostSize_64f_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiAverageError\_64f\_C4R*.

## NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff\_Inf primitives.

- **NppStatus** `nppiMaximumRelativeErrorGetBufferHostSize_8u_C1R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiMaximumRelativeError\_8u\_C1R*.
- **NppStatus** `nppiMaximumRelativeErrorGetBufferHostSize_8s_C1R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size for *nppiMaximumRelativeError\_8s\_C1R*.

- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_16u\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_16s\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16sc\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_16sc\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32u\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_32u\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32s\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_32s\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32sc\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_32sc\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_32f\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32fc\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_32fc\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_64f\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_64f\\_C1R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_8u\\_C2R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_8s\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_8s\\_C2R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMaximumRelativeError\\_16u\\_C2R](#).*
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))

*Buffer size for `nppiMaximumRelativeError_16s_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C2R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_16sc_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32u_C2R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_32u_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32s_C2R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_32s_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32sc_C2R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_32sc_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32f_C2R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_32f_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_32fc_C2R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_32fc_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_64f_C2R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_64f_C2R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8u_C3R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_8u_C3R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_8s_C3R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_8s_C3R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16u_C3R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_16u_C3R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16s_C3R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_16s_C3R`.*

- `NppStatus nppiMaximumRelativeErrorGetBufferHostSize_16sc_C3R (NppiSize oSizeROI, int *hpBufferSize)`

*Buffer size for `nppiMaximumRelativeError_16sc_C3R`.*

- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32u\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32u\_C3R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32s\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32s\_C3R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32sc\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32sc\_C3R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32f\_C3R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32fc\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32fc\_C3R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_64f\\_C3R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_64f\_C3R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_8u\_C4R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_8s\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_8s\_C4R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_16u\_C4R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_16s\_C4R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_16sc\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_16sc\_C4R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32u\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32u\_C4R*.
- [NppStatus nppiMaximumRelativeErrorGetBufferHostSize\\_32s\\_C4R](#) (NppiSize oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32s\_C4R*.



- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize\_32sc\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32sc\_C4R*.
- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32f\_C4R*.
- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize\_32fc\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_32fc\_C4R*.
- **NppStatus** **nppiMaximumRelativeErrorGetBufferHostSize\_64f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMaximumRelativeError\_64f\_C4R*.

## NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff\_Inf primitives.

- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiAverageRelativeError\_8u\_C1R*.
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize\_8s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiAverageRelativeError\_8s\_C1R*.
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiAverageRelativeError\_16u\_C1R*.
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiAverageRelativeError\_16s\_C1R*.
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize\_16sc\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiAverageRelativeError\_16sc\_C1R*.
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize\_32u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiAverageRelativeError\_32u\_C1R*.
- **NppStatus** **nppiAverageRelativeErrorGetBufferHostSize\_32s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiAverageRelativeError\_32s\_C1R*.

- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_32sc\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_32sc\\_C1R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_32f\\_C1R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_32fc\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_32fc\\_C1R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_64f\\_C1R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_64f\\_C1R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_8u\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_8s\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_8s\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_16u\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_16s\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_16sc\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_16sc\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_32u\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_32u\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_32s\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_32s\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_32sc\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiAverageRelativeError\\_32sc\\_C2R](#).*
- [NppStatus nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C2R](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))

*Buffer size for `nppiAverageRelativeError_32f_C2R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_32fc_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_32fc_C2R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_64f_C2R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_64f_C2R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_8u_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_8u_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_8s_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_8s_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_16u_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_16s_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_16sc_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_16sc_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_32u_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_32u_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_32s_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_32s_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_32sc_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_32sc_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_32f_C3R`.*

- `NppStatus` `nppiAverageRelativeErrorGetBufferHostSize_32fc_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size for `nppiAverageRelativeError_32fc_C3R`.*

- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_64f\\_C3R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_8u\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_8s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_8s\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_16u\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_16s\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_16sc\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_16sc\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_32u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_32u\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_32s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_32s\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_32sc\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_32sc\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_32f\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_32fc\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_32fc\\_C4R](#).*
- [NppStatus](#) [nppiAverageRelativeErrorGetBufferHostSize\\_64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiAverageRelativeError\\_64f\\_C4R](#).*

### 7.87.1 Detailed Description

Primitives for computing the statistical properties of an image.

Some statistical primitives also require scratch buffer during the computation. For details, please refer to [Scratch Buffer and Host Pointer](#).

### 7.87.2 Function Documentation

#### 7.87.2.1 NppStatus nppiAverageErrorGetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageError\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.2 NppStatus nppiAverageErrorGetBufferHostSize\_16s\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageError\\_16s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.3 NppStatus nppiAverageErrorGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageError\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.4 NppStatus nppiAverageErrorGetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageError\\_16s\\_C4R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.5 NppStatus nppiAverageErrorGetBufferHostSize\_16sc\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageError\\_16sc\\_C1R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.6 NppStatus nppiAverageErrorGetBufferHostSize\_16sc\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageError\\_16sc\\_C2R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.7 NppStatus nppiAverageErrorGetBufferHostSize\_16sc\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageError\\_16sc\\_C3R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.8 NppStatus nppiAverageErrorGetBufferHostSize\_16sc\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_16sc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.9 NppStatus nppiAverageErrorGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.10 NppStatus nppiAverageErrorGetBufferHostSize\_16u\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_16u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.11 NppStatus nppiAverageErrorGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.12 NppStatus nppiAverageErrorGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.13 NppStatus nppiAverageErrorGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.14 NppStatus nppiAverageErrorGetBufferHostSize\_32f\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32f\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).



**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.15 NppStatus nppiAverageErrorGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32f\\_C3R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.16 NppStatus nppiAverageErrorGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32f\\_C4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.17 NppStatus nppiAverageErrorGetBufferHostSize\_32fc\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32fc\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.18 NppStatus nppiAverageErrorGetBufferHostSize\_32fc\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32fc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.19 NppStatus nppiAverageErrorGetBufferHostSize\_32fc\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32fc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.20 NppStatus nppiAverageErrorGetBufferHostSize\_32fc\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32fc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.21 NppStatus nppiAverageErrorGetBufferHostSize\_32s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.22 NppStatus nppiAverageErrorGetBufferHostSize\_32s\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.23 NppStatus nppiAverageErrorGetBufferHostSize\_32s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.24 NppStatus nppiAverageErrorGetBufferHostSize\_32s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.25 NppStatus nppiAverageErrorGetBufferHostSize\_32sc\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32sc\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.26 NppStatus nppiAverageErrorGetBufferHostSize\_32sc\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32sc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.27 NppStatus nppiAverageErrorGetBufferHostSize\_32sc\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32sc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.28 NppStatus nppiAverageErrorGetBufferHostSize\_32sc\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32sc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.29 NppStatus nppiAverageErrorGetBufferHostSize\_32u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32u\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.30 NppStatus nppiAverageErrorGetBufferHostSize\_32u\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32u\\_C2R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.31 NppStatus nppiAverageErrorGetBufferHostSize\_32u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_32u\\_C3R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.32 NppStatus nppiAverageErrorGetBufferHostSize\_32u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_32u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.33 NppStatus nppiAverageErrorGetBufferHostSize\_64f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_64f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.34 NppStatus nppiAverageErrorGetBufferHostSize\_64f\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_64f\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.35 NppStatus nppiAverageErrorGetBufferHostSize\_64f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_64f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.36 NppStatus nppiAverageErrorGetBufferHostSize\_64f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_64f\\_C4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.37 NppStatus nppiAverageErrorGetBufferHostSize\_8s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_8s\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.38 NppStatus nppiAverageErrorGetBufferHostSize\_8s\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_8s\\_C2R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.39 NppStatus nppiAverageErrorGetBufferHostSize\_8s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_8s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.40 NppStatus nppiAverageErrorGetBufferHostSize\_8s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_8s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.41 NppStatus nppiAverageErrorGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.42 NppStatus nppiAverageErrorGetBufferHostSize\_8u\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageError\\_8u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.43 NppStatus nppiAverageErrorGetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.44 NppStatus nppiAverageErrorGetBufferHostSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageError\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.45 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.46 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16s\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_16s\\_C2R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.47 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_16s\\_C3R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.48 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_16s\\_C4R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.87.2.49 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16sc\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_16sc\\_C1R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.50 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16sc\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_16sc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.51 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16sc\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_16sc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.52 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_16sc\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_16sc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.53 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C1R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError\\_16u\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.54 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C2R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError\\_16u\\_C2R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.55 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C3R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError\\_16u\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.56 `NppStatus nppiAverageRelativeErrorGetBufferHostSize_16u_C4R` (`NppiSize oSizeROI`, `int * hpBufferSize`)

Buffer size for [nppiAverageRelativeError\\_16u\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.57 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.58 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32f\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32f\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.59 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.60 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.61 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32fc\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_32fc\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.62 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32fc\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_32fc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.63 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32fc\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_32fc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.64 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32fc\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32fc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.65 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.66 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32s\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.67 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_32s\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.68 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_32s\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.69 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32sc\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_32sc\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.70 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32sc\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_32sc\\_C2R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.71 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32sc\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32sc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.72 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32sc\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32sc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.73 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_32u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.74 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32u\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_32u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.75 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_32u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.76 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_32u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_32u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.77 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_64f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_64f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.78 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_64f\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_64f\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.79 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_64f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_64f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.80 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_64f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_64f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.81 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_8s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.82 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8s\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_8s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.83 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_8s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.84 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiAverageRelativeError\\_8s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.85 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.86 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8u\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_8u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.87 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiAverageRelativeError\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.88 NppStatus nppiAverageRelativeErrorGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiAverageRelativeError\\_8u\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.89 NppStatus nppiMaximumErrorGetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_16s\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.90 NppStatus nppiMaximumErrorGetBufferHostSize\_16s\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_16s\\_C2R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.91 NppStatus nppiMaximumErrorGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_16s\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.92 NppStatus nppiMaximumErrorGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_16s\\_C4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.93 NppStatus nppiMaximumErrorGetBufferHostSize\_16sc\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_16sc\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.94 NppStatus nppiMaximumErrorGetBufferHostSize\_16sc\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_16sc\\_C2R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.95 NppStatus nppiMaximumErrorGetBufferHostSize\_16sc\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_16sc\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.96 NppStatus nppiMaximumErrorGetBufferHostSize\_16sc\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_16sc\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.97 NppStatus nppiMaximumErrorGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_16u\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.98 NppStatus nppiMaximumErrorGetBufferHostSize\_16u\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_16u\\_C2R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.99 NppStatus nppiMaximumErrorGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.100 NppStatus nppiMaximumErrorGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.101 NppStatus nppiMaximumErrorGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.102 NppStatus nppiMaximumErrorGetBufferHostSize\_32f\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32f\\_C2R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.103 NppStatus nppiMaximumErrorGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32f\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.104 NppStatus nppiMaximumErrorGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32f\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.105 NppStatus nppiMaximumErrorGetBufferHostSize\_32fc\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32fc\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.106 NppStatus nppiMaximumErrorGetBufferHostSize\_32fc\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32fc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.107 NppStatus nppiMaximumErrorGetBufferHostSize\_32fc\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32fc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.108 NppStatus nppiMaximumErrorGetBufferHostSize\_32fc\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32fc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.109 NppStatus nppiMaximumErrorGetBufferHostSize\_32s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.110 NppStatus nppiMaximumErrorGetBufferHostSize\_32s\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.111 NppStatus nppiMaximumErrorGetBufferHostSize\_32s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.112 NppStatus nppiMaximumErrorGetBufferHostSize\_32s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.113 NppStatus nppiMaximumErrorGetBufferHostSize\_32sc\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32sc\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.114 NppStatus nppiMaximumErrorGetBufferHostSize\_32sc\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32sc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.115 NppStatus nppiMaximumErrorGetBufferHostSize\_32sc\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32sc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.116 NppStatus nppiMaximumErrorGetBufferHostSize\_32sc\_C4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32sc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.117 NppStatus nppiMaximumErrorGetBufferHostSize\_32u\_C1R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.118 NppStatus nppiMaximumErrorGetBufferHostSize\_32u\_C2R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.119 NppStatus nppiMaximumErrorGetBufferHostSize\_32u\_C3R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_32u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.120 NppStatus nppiMaximumErrorGetBufferHostSize\_32u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_32u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.121 NppStatus nppiMaximumErrorGetBufferHostSize\_64f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_64f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.122 NppStatus nppiMaximumErrorGetBufferHostSize\_64f\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMaximumError\\_64f\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.123 **NppStatus nppiMaximumErrorGetBufferHostSize\_64f\_C3R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_64f\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.124 **NppStatus nppiMaximumErrorGetBufferHostSize\_64f\_C4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_64f\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.125 **NppStatus nppiMaximumErrorGetBufferHostSize\_8s\_C1R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_8s\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.126 **NppStatus nppiMaximumErrorGetBufferHostSize\_8s\_C2R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_8s\\_C2R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.127 NppStatus nppiMaximumErrorGetBufferHostSize\_8s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_8s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.128 NppStatus nppiMaximumErrorGetBufferHostSize\_8s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_8s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.129 NppStatus nppiMaximumErrorGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumError\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.130 NppStatus nppiMaximumErrorGetBufferHostSize\_8u\_C2R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_8u\\_C2R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.131 NppStatus nppiMaximumErrorGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_8u\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.132 NppStatus nppiMaximumErrorGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumError\\_8u\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.87.2.133 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMaximumRelativeError\\_16s\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.134 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16s\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.135 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.136 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.137 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16sc\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16sc\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.138 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16sc\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16sc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.139 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16sc\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16sc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.140 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16sc\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16sc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.141 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.142 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16u\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.143 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.144 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.145 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.146 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32f\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32f\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.147 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.148 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.149 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32fc\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32fc\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.150 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32fc\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32fc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.151 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32fc\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32fc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.152 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32fc\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32fc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.153 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.154 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32s\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.155 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.156 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.157 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32sc\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32sc\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.158 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32sc\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32sc\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.159 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32sc\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32sc\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.160 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32sc\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32sc\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.161 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.162 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32u\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.163 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.164 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_32u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_32u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.165 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_64f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_64f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.166 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_64f\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_64f\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.167 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_64f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_64f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.168 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_64f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_64f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.169 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.170 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8s\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8s\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.171 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.172 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.173 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.174 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8u\_C2R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8u\\_C2R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.175 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.87.2.176 NppStatus nppiMaximumRelativeErrorGetBufferHostSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMaximumRelativeError\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.88 Sum

Primitives for computing the sum of all the pixel values in an image.

### Sum

Given an image  $pSrc$  with width  $W$  and height  $H$ , the sum will be computed as

$$Sum = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

All the results are stored in a 64-bit double precision format, except for two primitives `npSum_8u64s_C1R` and `npSum_8u64s_C4R`.

The sum functions require additional scratch buffer for computations.

- `NppStatus npSum_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pSum)  
*One-channel 8-bit unsigned image sum.*
- `NppStatus npSum_8u64s_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64s` \*pSum)  
*One-channel 8-bit unsigned image sum.*
- `NppStatus npSum_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pSum)  
*One-channel 16-bit unsigned image sum.*
- `NppStatus npSum_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pSum)  
*One-channel 16-bit signed image sum.*
- `NppStatus npSum_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pSum)  
*One-channel 32-bit floating point image sum.*
- `NppStatus npSum_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aSum[3])  
*Three-channel 8-bit unsigned image sum.*
- `NppStatus npSum_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aSum[3])  
*Three-channel 16-bit unsigned image sum.*
- `NppStatus npSum_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aSum[3])  
*Three-channel 16-bit signed image sum.*
- `NppStatus npSum_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aSum[3])  
*Three-channel 32-bit floating point image sum.*



- **NppStatus** **nppiSum\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*Four-channel 8-bit unsigned image sum ignoring alpha channel.*
- **NppStatus** **nppiSum\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*Four-channel 16-bit unsigned image sum ignoring alpha channel.*
- **NppStatus** **nppiSum\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*Four-channel 16-bit signed image sum ignoring alpha channel.*
- **NppStatus** **nppiSum\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[3])  
*Four-channel 32-bit floating point image sum ignoring alpha channel.*
- **NppStatus** **nppiSum\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[4])  
*Four-channel 8-bit unsigned image sum.*
- **NppStatus** **nppiSum\_8u64s\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64s** aSum[4])  
*Four-channel 8-bit unsigned image sum.*
- **NppStatus** **nppiSum\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[4])  
*Four-channel 16-bit unsigned image sum.*
- **NppStatus** **nppiSum\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[4])  
*Four-channel 16-bit signed image sum.*
- **NppStatus** **nppiSum\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp64f** aSum[4])  
*Four-channel 32-bit floating point image sum.*

## SumGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the sum primitives.

- **NppStatus** **nppiSumGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiSum\_8u\_C1R**.*
- **NppStatus** **nppiSumGetBufferHostSize\_8u64s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiSum\_8u64s\_C1R**.*
- **NppStatus** **nppiSumGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiSum\_16u\_C1R**.*

- [NppStatus nppiSumGetBufferHostSize\\_16s\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_16s\\_C1R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_32f\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_32f\\_C1R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_8u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_8u\\_C3R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_16u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_16u\\_C3R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_16s\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_16s\\_C3R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_32f\\_C3R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_8u\\_AC4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_16u\\_AC4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_16s\\_AC4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_32f\\_AC4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_8u64s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_8u64s\\_C4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_8u\\_C4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_16u\\_C4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_16s\\_C4R](#).*
- [NppStatus nppiSumGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiSum\\_32f\\_C4R](#).*

### 7.88.1 Detailed Description

Primitives for computing the sum of all the pixel values in an image.

## 7.88.2 Function Documentation

### 7.88.2.1 `NppStatus nppiSum_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

Four-channel 16-bit signed image sum ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16s\\_AC4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.2 `NppStatus nppiSum_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pSum)`

One-channel 16-bit signed image sum.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

*pSum* Pointer to the computed sum.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.3 `NppStatus nppiSum_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aSum[3])`

Three-channel 16-bit signed image sum.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16s\\_C3R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.88.2.4 NppStatus nppiSum\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[4])**

Four-channel 16-bit signed image sum.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_16s\\_C4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.88.2.5 NppStatus nppiSum\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[3])**

Four-channel 16-bit unsigned image sum ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.88.2.6 NppStatus nppiSum\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pSum)**

One-channel 16-bit unsigned image sum.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pSum* Pointer to the computed sum.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.88.2.7 NppStatus nppiSum\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[3])

Three-channel 16-bit unsigned image sum.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSumGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.88.2.8 NppStatus nppiSum\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aSum[4])

Four-channel 16-bit unsigned image sum.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use  
[nppiSumGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.9 **NppStatus nppiSum\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aSum*[3])

Four-channel 32-bit floating point image sum ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.10 **NppStatus nppiSum\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pSum*)

One-channel 32-bit floating point image sum.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pSum* Pointer to the computed sum.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.11 **NppStatus nppiSum\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aSum*[3])

Three-channel 32-bit floating point image sum.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.12 **NppStatus nppiSum\_32f\_C4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aSum*[4])

Four-channel 32-bit floating point image sum.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_32f\\_C4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.13 **NppStatus nppiSum\_8u64s\_C1R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64s \* *pSum*)

One-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u64s\\_C1R](#) to determine the minium number of bytes required.

*pSum* Pointer to the computed sum.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.14 **NppStatus nppiSum\_8u64s\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64s *aSum*[4])

Four-channel 8-bit unsigned image sum.

The result is 64-bit long long integer.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiSumGetBufferHostSize\\_8u64s\\_C4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.88.2.15** `NppStatus nppiSum_8u_AC4R (const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u *pDeviceBuffer, Npp64f aSum[3])`

Four-channel 8-bit unsigned image sum ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u\\_AC4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel (alpha channel is not computed).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.88.2.16** `NppStatus nppiSum_8u_C1R (const Npp8u *pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u *pDeviceBuffer, Npp64f *pSum)`

One-channel 8-bit unsigned image sum.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSumGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

*pSum* Pointer to the computed sum.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.88.2.17 NppStatus nppiSum\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aSum*[3])

Three-channel 8-bit unsigned image sum.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiSumGetBufferHostSize\\_8u\\_C3R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.18 NppStatus nppiSum\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aSum*[4])

Four-channel 8-bit unsigned image sum.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use

[nppiSumGetBufferHostSize\\_8u\\_C4R](#) to determine the minium number of bytes required.

*aSum* Array that contains computed sum for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.88.2.19 NppStatus nppiSumGetBufferHostSize\_16s\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_16s\\_AC4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.88.2.20 NppStatus nppiSumGetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_16s\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.88.2.21 NppStatus nppiSumGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_16s\\_C3R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.88.2.22 NppStatus nppiSumGetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_16s\\_C4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.88.2.23 NppStatus nppiSumGetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_16u\\_AC4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.24 NppStatus nppiSumGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiSum\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.25 NppStatus nppiSumGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiSum\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.26 NppStatus nppiSumGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiSum\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.27 NppStatus nppiSumGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiSum\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.28 NppStatus nppiSumGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiSum\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.29 NppStatus nppiSumGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiSum\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.30 NppStatus nppiSumGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiSum\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.31 NppStatus nppiSumGetBufferHostSize\_8u64s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiSum\\_8u64s\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.32 NppStatus nppiSumGetBufferHostSize\_8u64s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiSum\\_8u64s\\_C4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.88.2.33 NppStatus nppiSumGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiSum\\_8u\\_AC4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

#### 7.88.2.34 NppStatus nppiSumGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_8u\\_C1R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.88.2.35 NppStatus nppiSumGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_8u\\_C3R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.88.2.36 NppStatus nppiSumGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiSum\\_8u\\_C4R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.89 Min

Primitives for computing the minimal pixel value of an image.

### Min

The scratch buffer is required by the min functions.

- `NppStatus nppiMin_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` \*pMin)  
*One-channel 8-bit unsigned image min.*
- `NppStatus nppiMin_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` \*pMin)  
*One-channel 16-bit unsigned image min.*
- `NppStatus nppiMin_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` \*pMin)  
*One-channel 16-bit signed image min.*
- `NppStatus nppiMin_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` \*pMin)  
*One-channel 32-bit floating point image min.*
- `NppStatus nppiMin_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` aMin[3])  
*Three-channel 8-bit unsigned image min.*
- `NppStatus nppiMin_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` aMin[3])  
*Three-channel 16-bit unsigned image min.*
- `NppStatus nppiMin_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` aMin[3])  
*Three-channel 16-bit signed image min.*
- `NppStatus nppiMin_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` aMin[3])  
*Three-channel 32-bit floating point image min.*
- `NppStatus nppiMin_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` aMin[4])  
*Four-channel 8-bit unsigned image min.*
- `NppStatus nppiMin_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` aMin[4])  
*Four-channel 16-bit unsigned image min.*
- `NppStatus nppiMin_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` aMin[4])

*Four-channel 16-bit signed image min.*

- `NppStatus nppiMin_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` aMin[4])

*Four-channel 32-bit floating point image min.*

- `NppStatus nppiMin_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` aMin[3])

*Four-channel 8-bit unsigned image min ignoring alpha channel.*

- `NppStatus nppiMin_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` aMin[3])

*Four-channel 16-bit unsigned image min ignoring alpha channel.*

- `NppStatus nppiMin_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` aMin[3])

*Four-channel 16-bit signed image min ignoring alpha channel.*

- `NppStatus nppiMin_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` aMin[3])

*Four-channel 32-bit floating point image min ignoring alpha channel.*

## MinGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the min primitives.

- `NppStatus nppiMinGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_8u_C1R`.*
- `NppStatus nppiMinGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_16u_C1R`.*
- `NppStatus nppiMinGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_16s_C1R`.*
- `NppStatus nppiMinGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_32f_C1R`.*
- `NppStatus nppiMinGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_8u_C3R`.*
- `NppStatus nppiMinGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_16u_C3R`.*
- `NppStatus nppiMinGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_16s_C3R`.*
- `NppStatus nppiMinGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMin_32f_C3R`.*



- [NppStatus nppiMinGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_8u\\_C4R](#).*
- [NppStatus nppiMinGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_16u\\_C4R](#).*
- [NppStatus nppiMinGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_16s\\_C4R](#).*
- [NppStatus nppiMinGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_32f\\_C4R](#).*
- [NppStatus nppiMinGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_8u\\_AC4R](#).*
- [NppStatus nppiMinGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_16u\\_AC4R](#).*
- [NppStatus nppiMinGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_16s\\_AC4R](#).*
- [NppStatus nppiMinGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMin\\_32f\\_AC4R](#).*

### 7.89.1 Detailed Description

Primitives for computing the minimal pixel value of an image.

### 7.89.2 Function Documentation

#### 7.89.2.1 [NppStatus nppiMin\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp16s](#) aMin[3])

Four-channel 16-bit signed image min ignoring alpha channel.

##### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_AC4R](#) to determine the minium number of bytes required.

[aMin](#) Array that contains the computed minimum results for each channel (alpha channel is not processed).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.2 NppStatus nppiMin\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s \* pMin)

One-channel 16-bit signed image min.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

*pMin* Pointer to the computed minimum result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.3 NppStatus nppiMin\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s aMin[3])

Three-channel 16-bit signed image min.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_C3R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.4 NppStatus nppiMin\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s aMin[4])

Four-channel 16-bit signed image min.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16s\\_C4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.5 NppStatus nppiMin\_16u\_AC4R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u *aMin*[3])

Four-channel 16-bit unsigned image min ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel (alpha channel is not processed).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.6 NppStatus nppiMin\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u \* *pMin*)

One-channel 16-bit unsigned image min.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pMin* Pointer to the computed minimum result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.7 NppStatus nppiMin\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u *aMin*[3])

Three-channel 16-bit unsigned image min.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.89.2.8 NppStatus nppiMin\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMin[4])**

Four-channel 16-bit unsigned image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.89.2.9 NppStatus nppiMin\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMin[3])**

Four-channel 32-bit floating point image min ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel (alpha channel is not processed).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.89.2.10 NppStatus nppiMin\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f \* pMin)**

One-channel 32-bit floating point image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pMin* Pointer to the computed minimum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.89.2.11** `NppStatus nppiMin_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[3])`

Three-channel 32-bit floating point image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.89.2.12** `NppStatus nppiMin_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMin[4])`

Four-channel 32-bit floating point image min.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_32f\\_C4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.13 `NppStatus nppiMin_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3])`

Four-channel 8-bit unsigned image min ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMinGetBufferHostSize_8u_AC4R` to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel (alpha channel is not processed).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.14 `NppStatus nppiMin_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMin)`

One-channel 8-bit unsigned image min.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMinGetBufferHostSize_8u_C1R` to determine the minium number of bytes required.

*pMin* Pointer to the computed minimum result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.89.2.15 `NppStatus nppiMin_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[3])`

Three-channel 8-bit unsigned image min.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMinGetBufferHostSize_8u_C3R` to determine the minium number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.89.2.16** `NppStatus nppiMin_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMin[4])`

Four-channel 8-bit unsigned image min.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinGetBufferHostSize\\_8u\\_C4R](#) to determine the minimum number of bytes required.

*aMin* Array that contains the computed minimum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.89.2.17** `NppStatus nppiMinGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMin\\_16s\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.18** `NppStatus nppiMinGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMin\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.19 NppStatus nppiMinGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiMin\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.20 NppStatus nppiMinGetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiMin\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.21 NppStatus nppiMinGetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiMin\\_16u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.22 NppStatus nppiMinGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiMin\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.23 NppStatus nppiMinGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMin\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.24 NppStatus nppiMinGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMin\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.25 NppStatus nppiMinGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMin\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.26 NppStatus nppiMinGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMin\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.27 NppStatus nppiMinGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMin\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.28 NppStatus nppiMinGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMin\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.29 NppStatus nppiMinGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMin\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.30 NppStatus nppiMinGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMin\\_8u\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.31 NppStatus nppiMinGetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMin\\_8u\\_C3R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.89.2.32 NppStatus nppiMinGetBufferHostSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMin\\_8u\\_C4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

## 7.90 MinIndx

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

### MinIndx

If there are several minima in the selected ROI, the function returns one on the top leftmost position.

The scratch buffer is required by the functions.

- **NppStatus nppiMinIndx\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** \*pMin, int \*pIndexX, int \*pIndexY)  
*One-channel 8-bit unsigned image MinIndx.*
- **NppStatus nppiMinIndx\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** \*pMin, int \*pIndexX, int \*pIndexY)  
*One-channel 16-bit unsigned image MinIndx.*
- **NppStatus nppiMinIndx\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** \*pMin, int \*pIndexX, int \*pIndexY)  
*One-channel 16-bit signed image MinIndx.*
- **NppStatus nppiMinIndx\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** \*pMin, int \*pIndexX, int \*pIndexY)  
*One-channel 32-bit floating point image MinIndx.*
- **NppStatus nppiMinIndx\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 8-bit unsigned image MinIndx.*
- **NppStatus nppiMinIndx\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 16-bit unsigned image MinIndx.*
- **NppStatus nppiMinIndx\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 16-bit signed image MinIndx.*
- **NppStatus nppiMinIndx\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMin[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 32-bit floating point image MinIndx.*
- **NppStatus nppiMinIndx\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 8-bit unsigned image MinIndx.*
- **NppStatus nppiMinIndx\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 16-bit unsigned image MinIndx.*

- **NppStatus nppiMinIndx\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 16-bit signed image MinIndx.*
- **NppStatus nppiMinIndx\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMin[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 32-bit floating point image MinIndx.*
- **NppStatus nppiMinIndx\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMin[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 8-bit unsigned image MinIndx ignoring alpha channel.*
- **NppStatus nppiMinIndx\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMin[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 16-bit unsigned image MinIndx ignoring alpha channel.*
- **NppStatus nppiMinIndx\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMin[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 16-bit signed image MinIndx ignoring alpha channel.*
- **NppStatus nppiMinIndx\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMin[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 32-bit floating point image MinIndx ignoring alpha channel.*

## MinIndxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinIndx primitives.

- **NppStatus nppiMinIndxGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C1R.*
- **NppStatus nppiMinIndxGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIndx\_16u\_C1R.*
- **NppStatus nppiMinIndxGetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIndx\_16s\_C1R.*
- **NppStatus nppiMinIndxGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIndx\_32f\_C1R.*
- **NppStatus nppiMinIndxGetBufferHostSize\_8u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C3R.*
- **NppStatus nppiMinIndxGetBufferHostSize\_16u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIndx\_16u\_C3R.*
- **NppStatus nppiMinIndxGetBufferHostSize\_16s\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIndx\_16s\_C3R.*

- [NppStatus nppiMinIdxGetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_32f\_C3R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_8u\_C4R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_16u\_C4R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_16s\_C4R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_32f\_C4R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_8u\_AC4R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_16u\_AC4R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_16s\_AC4R.*
- [NppStatus nppiMinIdxGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMinIdx\_32f\_AC4R.*

## 7.90.1 Detailed Description

Primitives for computing the minimal value and its indices (X and Y coordinates) of an image.

## 7.90.2 Function Documentation

### 7.90.2.1 [NppStatus nppiMinIdx\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp16s](#) aMin[3], int aIndexX[3], int aIndexY[3])

Four-channel 16-bit signed image MinIdx ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_16s\\_AC4R](#) to determine the minimum number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.90.2.2 NppStatus nppiMinIndx\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s \* pMin, int \* pIndexX, int \* pIndexY)**

One-channel 16-bit signed image MinIndx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

*pMin* Pointer to the computed min result.

*pIndexX* Pointer to the X coordinate of the image min value.

*pIndexY* Pointer to the Y coordinate of the image min value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.90.2.3 NppStatus nppiMinIndx\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16s aMin[3], int aIndexX[3], int aIndexY[3])**

Three-channel 16-bit signed image MinIndx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16s\\_C3R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.4 NppStatus nppiMinIndx\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16s *aMin*[4], int *aIndexX*[4], int *aIndexY*[4])

Four-channel 16-bit signed image MinIndx.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16s\\_C4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.5 NppStatus nppiMinIndx\_16u\_AC4R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Four-channel 16-bit unsigned image MinIndx ignoring alpha channel.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.6 NppStatus nppiMinIndx\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u \* *pMin*, int \* *pIndexX*, int \* *pIndexY*)

One-channel 16-bit unsigned image MinIndx.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.



*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pMin* Pointer to the computed min result.

*pIndexX* Pointer to the X coordinate of the image min value.

*pIndexY* Pointer to the Y coordinate of the image min value.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.7 NppStatus nppiMinIndx\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMin[3], int aIndexX[3], int aIndexY[3])

Three-channel 16-bit unsigned image MinIndx.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.8 NppStatus nppiMinIndx\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMin[4], int aIndexX[4], int aIndexY[4])

Four-channel 16-bit unsigned image MinIndx.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.90.2.9 NppStatus nppiMinIndx\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp32f *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Four-channel 32-bit floating point image MinIndx ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.90.2.10 NppStatus nppiMinIndx\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp32f \* *pMin*, int \* *pIndexX*, int \* *pIndexY*)

One-channel 32-bit floating point image MinIndx.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pMin* Pointer to the computed min result.

*pIndexX* Pointer to the X coordinate of the image min value.

*pIndexY* Pointer to the Y coordinate of the image min value.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.90.2.11 NppStatus nppiMinIndx\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp32f *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Three-channel 32-bit floating point image MinIndx.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.90.2.12 NppStatus nppiMinIndx\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMin[4], int aIndexX[4], int aIndexY[4])**

Four-channel 32-bit floating point image MinIndx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_32f\\_C4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.90.2.13 NppStatus nppiMinIndx\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u aMin[3], int aIndexX[3], int aIndexY[3])**

Four-channel 8-bit unsigned image MinIndx ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_8u\\_AC4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.14 NppStatus nppiMinIdx\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp8u \* *pMin*, int \* *pIndexX*, int \* *pIndexY*)

One-channel 8-bit unsigned image MinIdx.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

*pMin* Pointer to the computed min result.

*pIndexX* Pointer to the X coordinate of the image min value.

*pIndexY* Ppointer to the Y coordinate of the image min value.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.15 NppStatus nppiMinIdx\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp8u *aMin*[3], int *aIndexX*[3], int *aIndexY*[3])

Three-channel 8-bit unsigned image MinIdx.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIdxGetBufferHostSize\\_8u\\_C3R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.90.2.16 NppStatus nppiMinIdx\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp8u *aMin*[4], int *aIndexX*[4], int *aIndexY*[4])

Four-channel 8-bit unsigned image MinIdx.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMinIndxGetBufferHostSize\\_8u\\_C4R](#) to determine the minium number of bytes required.

*aMin* Array that contains the min values.

*aIndexX* Array that contains the X coordinates of the image min values.

*aIndexY* Array that contains the Y coordinates of the image min values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.90.2.17 NppStatus nppiMinIndxGetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the dvice scratch buffer size (in bytes) for nppiMinIndx\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.18 NppStatus nppiMinIndxGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the dvice scratch buffer size (in bytes) for nppiMinIndx\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.19 NppStatus nppiMinIndxGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the dvice scratch buffer size (in bytes) for nppiMinIndx\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.20 NppStatus nppiMinIdxGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMinIdx\_16s\_C4R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.21 NppStatus nppiMinIdxGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMinIdx\_8u\_AC4R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.22 NppStatus nppiMinIdxGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMinIdx\_16u\_C1R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.23 NppStatus nppiMinIndxGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx\_16u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.24 NppStatus nppiMinIndxGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.25 NppStatus nppiMinIndxGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.26 NppStatus nppiMinIndxGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.27 NppStatus nppiMinIdxGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiMinIdx\_32f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.28 NppStatus nppiMinIdxGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiMinIdx\_32f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.29 NppStatus nppiMinIdxGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiMinIdx\_8u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.90.2.30 NppStatus nppiMinIndxGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.31 NppStatus nppiMinIndxGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.90.2.32 NppStatus nppiMinIndxGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMinIndx\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.91 Max

Primitives for computing the maximal pixel value of an image.

### Max

The scratch buffer is required by the functions.

- `NppStatus nppiMax_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` \*pMax)  
*One-channel 8-bit unsigned image Max.*
- `NppStatus nppiMax_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` \*pMax)  
*One-channel 16-bit unsigned image Max.*
- `NppStatus nppiMax_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` \*pMax)  
*One-channel 16-bit signed image Max.*
- `NppStatus nppiMax_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` \*pMax)  
*One-channel 32-bit floating point image Max.*
- `NppStatus nppiMax_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` aMax[3])  
*Three-channel 8-bit unsigned image Max.*
- `NppStatus nppiMax_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` aMax[3])  
*Three-channel 16-bit unsigned image Max.*
- `NppStatus nppiMax_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` aMax[3])  
*Three-channel 16-bit signed image Max.*
- `NppStatus nppiMax_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` aMax[3])  
*Three-channel 32-bit floating point image Max.*
- `NppStatus nppiMax_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` aMax[4])  
*Four-channel 8-bit unsigned image Max.*
- `NppStatus nppiMax_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` aMax[4])  
*Four-channel 16-bit unsigned image Max.*
- `NppStatus nppiMax_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` aMax[4])

*Four-channel 16-bit signed image Max.*

- `NppStatus nppiMax_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` aMax[4])

*Four-channel 32-bit floating point image Max.*

- `NppStatus nppiMax_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp8u` aMax[3])

*Four-channel 8-bit unsigned image Max ignoring alpha channel.*

- `NppStatus nppiMax_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16u` aMax[3])

*Four-channel 16-bit unsigned image Max ignoring alpha channel.*

- `NppStatus nppiMax_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp16s` aMax[3])

*Four-channel 16-bit signed image Max ignoring alpha channel.*

- `NppStatus nppiMax_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp32f` aMax[3])

*Four-channel 32-bit floating point image Max ignoring alpha channel.*

## MaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Max primitives.

- `NppStatus nppiMaxGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_8u_C1R`.*
- `NppStatus nppiMaxGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_16u_C1R`.*
- `NppStatus nppiMaxGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_16s_C1R`.*
- `NppStatus nppiMaxGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_32f_C1R`.*
- `NppStatus nppiMaxGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_8u_C3R`.*
- `NppStatus nppiMaxGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_16u_C3R`.*
- `NppStatus nppiMaxGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_16s_C3R`.*
- `NppStatus nppiMaxGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMax_32f_C3R`.*

- [NppStatus nppiMaxGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_8u\\_C4R](#).*
- [NppStatus nppiMaxGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_16u\\_C4R](#).*
- [NppStatus nppiMaxGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_16s\\_C4R](#).*
- [NppStatus nppiMaxGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_32f\\_C4R](#).*
- [NppStatus nppiMaxGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_8u\\_AC4R](#).*
- [NppStatus nppiMaxGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_16u\\_AC4R](#).*
- [NppStatus nppiMaxGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_16s\\_AC4R](#).*
- [NppStatus nppiMaxGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMax\\_32f\\_AC4R](#).*

### 7.91.1 Detailed Description

Primitives for computing the maximal pixel value of an image.

### 7.91.2 Function Documentation

#### 7.91.2.1 [NppStatus nppiMax\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp16s](#) aMax[3])

Four-channel 16-bit signed image Max ignoring alpha channel.

#### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_AC4R](#) to determine the maximum number of bytes required.

[aMax](#) Array that contains the computed maximum results for each channel (alpha channel is not processed).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.2 NppStatus nppiMax\_16s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16s \* *pMax*)

One-channel 16-bit signed image Max.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed maximum result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.3 NppStatus nppiMax\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16s *aMax*[3])

Three-channel 16-bit signed image Max.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.4 NppStatus nppiMax\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16s *aMax*[4])

Four-channel 16-bit signed image Max.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16s\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.5 NppStatus nppiMax\_16u\_AC4R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u *aMax*[3])

Four-channel 16-bit unsigned image Max ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_AC4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel (alpha channel is not processed).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.6 NppStatus nppiMax\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u \* *pMax*)

One-channel 16-bit unsigned image Max.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed maximum result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.7 NppStatus nppiMax\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u *aMax*[3])

Three-channel 16-bit unsigned image Max.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.8 NppStatus nppiMax\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp16u aMax[4])**

Four-channel 16-bit unsigned image Max.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_16u\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.9 NppStatus nppiMax\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMax[3])**

Four-channel 32-bit floating point image Max ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_AC4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel (alpha channel is not processed).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.10 NppStatus nppiMax\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f \* pMax)**

One-channel 32-bit floating point image Max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed maximum result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.11** `NppStatus nppiMax_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[3])`

Three-channel 32-bit floating point image Max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.12** `NppStatus nppiMax_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp32f aMax[4])`

Four-channel 32-bit floating point image Max.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_32f\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.91.2.13 `NppStatus nppiMax_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3])`

Four-channel 8-bit unsigned image Max ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_8u\\_AC4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel (alpha channel is not processed).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.14 `NppStatus nppiMax_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u * pMax)`

One-channel 8-bit unsigned image Max.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_8u\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed maximum result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.91.2.15 `NppStatus nppiMax_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[3])`

Three-channel 8-bit unsigned image Max.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_8u\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.16** `NppStatus nppiMax_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp8u aMax[4])`

Four-channel 8-bit unsigned image Max.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxGetBufferHostSize\\_8u\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the computed maximum results for each channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.91.2.17** `NppStatus nppiMaxGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMax\\_16s\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.18** `NppStatus nppiMaxGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMax\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.19 NppStatus nppiMaxGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.20 NppStatus nppiMaxGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.21 NppStatus nppiMaxGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_16u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.22 NppStatus nppiMaxGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.23 NppStatus nppiMaxGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.24 NppStatus nppiMaxGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.25 NppStatus nppiMaxGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.26 NppStatus nppiMaxGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.27 NppStatus nppiMaxGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.28 NppStatus nppiMaxGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.29 NppStatus nppiMaxGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.30 NppStatus nppiMaxGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.31 NppStatus nppiMaxGetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.91.2.32 NppStatus nppiMaxGetBufferHostSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMax\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.92 MaxIdx

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

### MaxIdx

If there are several maxima in the selected region of interest, the function returns one on the top leftmost position.

The scratch buffer is required by the functions.

- **NppStatus nppiMaxIdx\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** \*pMax, int \*pIndexX, int \*pIndexY)  
*One-channel 8-bit unsigned image MaxIdx.*
- **NppStatus nppiMaxIdx\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** \*pMax, int \*pIndexX, int \*pIndexY)  
*One-channel 16-bit unsigned image MaxIdx.*
- **NppStatus nppiMaxIdx\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** \*pMax, int \*pIndexX, int \*pIndexY)  
*One-channel 16-bit signed image MaxIdx.*
- **NppStatus nppiMaxIdx\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** \*pMax, int \*pIndexX, int \*pIndexY)  
*One-channel 32-bit floating point image MaxIdx.*
- **NppStatus nppiMaxIdx\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMax[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 8-bit unsigned image MaxIdx.*
- **NppStatus nppiMaxIdx\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMax[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 16-bit unsigned image MaxIdx.*
- **NppStatus nppiMaxIdx\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 16-bit signed image MaxIdx.*
- **NppStatus nppiMaxIdx\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMax[3], int aIndexX[3], int aIndexY[3])  
*Three-channel 32-bit floating point image MaxIdx.*
- **NppStatus nppiMaxIdx\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMax[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 8-bit unsigned image MaxIdx.*
- **NppStatus nppiMaxIdx\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMax[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 16-bit unsigned image MaxIdx.*

- **NppStatus nppiMaxIdx\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 16-bit signed image MaxIdx.*
- **NppStatus nppiMaxIdx\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMax[4], int aIndexX[4], int aIndexY[4])  
*Four-channel 32-bit floating point image MaxIdx.*
- **NppStatus nppiMaxIdx\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp8u** aMax[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 8-bit unsigned image MaxIdx ignoring alpha channel.*
- **NppStatus nppiMaxIdx\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16u** aMax[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 16-bit unsigned image MaxIdx ignoring alpha channel.*
- **NppStatus nppiMaxIdx\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp16s** aMax[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 16-bit signed image MaxIdx ignoring alpha channel.*
- **NppStatus nppiMaxIdx\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp8u** \*pDeviceBuffer, **Npp32f** aMax[3], int aIndexX[3], int aIndexY[3])  
*Four-channel 32-bit floating point image MaxIdx ignoring alpha channel.*

## MaxIdxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MaxIdx primitives.

- **NppStatus nppiMaxIdxGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIdx\_8u\_C1R.*
- **NppStatus nppiMaxIdxGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIdx\_16u\_C1R.*
- **NppStatus nppiMaxIdxGetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIdx\_16s\_C1R.*
- **NppStatus nppiMaxIdxGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIdx\_32f\_C1R.*
- **NppStatus nppiMaxIdxGetBufferHostSize\_8u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIdx\_8u\_C3R.*
- **NppStatus nppiMaxIdxGetBufferHostSize\_16u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIdx\_16u\_C3R.*
- **NppStatus nppiMaxIdxGetBufferHostSize\_16s\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIdx\_16s\_C3R.*



- [NppStatus nppiMaxIndxGetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C3R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_AC4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_AC4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_AC4R.*
- [NppStatus nppiMaxIndxGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_AC4R.*

### 7.92.1 Detailed Description

Primitives for computing the maximal value and its indices (X and Y coordinates) of an image.

### 7.92.2 Function Documentation

#### 7.92.2.1 [NppStatus nppiMaxIndx\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp16s](#) aMax[3], int aIndexX[3], int aIndexY[3])

Four-channel 16-bit signed image MaxIndx ignoring alpha channel.

##### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMaxIndxGetBufferHostSize\\_16s\\_AC4R](#) to determine the maximum number of bytes required.

[aMax](#) Array that contains the max values.

[aIndexX](#) Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.92.2.2** `NppStatus nppiMaxIdx_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s * pMax, int * pIndexX, int * pIndexY)`

One-channel 16-bit signed image MaxIdx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16s\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed max result.

*pIndexX* Pointer to the X coordinate of the image max value.

*pIndexY* Pointer to the Y coordinate of the image max value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.92.2.3** `NppStatus nppiMaxIdx_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16s aMax[3], int aIndexX[3], int aIndexY[3])`

Three-channel 16-bit signed image MaxIdx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16s\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.92.2.4 NppStatus nppiMaxIndx\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16s *aMax*[4], int *aIndexX*[4], int *aIndexY*[4])

Four-channel 16-bit signed image MaxIndx.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_16s\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.92.2.5 NppStatus nppiMaxIndx\_16u\_AC4R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u *aMax*[3], int *aIndexX*[3], int *aIndexY*[3])

Four-channel 16-bit unsigned image MaxIndx ignoring alpha channel.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_16u\\_AC4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.92.2.6 NppStatus nppiMaxIndx\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp16u \* *pMax*, int \* *pIndexX*, int \* *pIndexY*)

One-channel 16-bit unsigned image MaxIndx.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16u\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed max result.

*pIndexX* Pointer to the X coordinate of the image max value.

*pIndexY* Pointer to the Y coordinate of the image max value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.92.2.7** `NppStatus nppiMaxIdx_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[3], int aIndexX[3], int aIndexY[3])`

Three-channel 16-bit unsigned image MaxIdx.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16u\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.92.2.8** `NppStatus nppiMaxIdx_16u_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp16u aMax[4], int aIndexX[4], int aIndexY[4])`

Four-channel 16-bit unsigned image MaxIdx.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_16u\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.92.2.9 NppStatus nppiMaxIndx\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp32f *aMax*[3], int *aIndexX*[3], int *aIndexY*[3])

Four-channel 32-bit floating point image MaxIndx ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_32f\\_AC4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.92.2.10 NppStatus nppiMaxIndx\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp32f \* *pMax*, int \* *pIndexX*, int \* *pIndexY*)

One-channel 32-bit floating point image MaxIndx.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_32f\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed max result.

*pIndexX* Pointer to the X coordinate of the image max value.

*pIndexY* Pointer to the Y coordinate of the image max value.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.92.2.11 NppStatus nppiMaxIndx\_32f\_C3R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp32f *aMax*[3], int *aIndexX*[3], int *aIndexY*[3])

Three-channel 32-bit floating point image MaxIndx.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_32f\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.92.2.12 NppStatus nppiMaxIdx\_32f\_C4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp32f aMax[4], int aIndexX[4], int aIndexY[4])**

Four-channel 32-bit floating point image MaxIdx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_32f\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.92.2.13 NppStatus nppiMaxIdx\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp8u aMax[3], int aIndexX[3], int aIndexY[3])**

Four-channel 8-bit unsigned image MaxIdx ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_8u\\_AC4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.92.2.14 NppStatus nppiMaxIdx\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp8u \* *pMax*, int \* *pIndexX*, int \* *pIndexY*)

One-channel 8-bit unsigned image MaxIdx.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_8u\\_C1R](#) to determine the maximum number of bytes required.

*pMax* Pointer to the computed max result.

*pIndexX* Pointer to the X coordinate of the image max value.

*pIndexY* Pointer to the Y coordinate of the image max value.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.92.2.15 NppStatus nppiMaxIdx\_8u\_C3R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp8u *aMax*[3], int *aIndexX*[3], int *aIndexY*[3])

Three-channel 8-bit unsigned image MaxIdx.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIdxGetBufferHostSize\\_8u\\_C3R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.92.2.16 NppStatus nppiMaxIdx\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp8u *aMax*[4], int *aIndexX*[4], int *aIndexY*[4])

Four-channel 8-bit unsigned image MaxIdx.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMaxIndxGetBufferHostSize\\_8u\\_C4R](#) to determine the maximum number of bytes required.

*aMax* Array that contains the max values.

*aIndexX* Array that contains the X coordinates of the image max values.

*aIndexY* Array that contains the Y coordinates of the image max values.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.92.2.17 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.18 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.19 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.20 NppStatus nppiMaxIndxGetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx\_16s\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.21 NppStatus nppiMaxIndxGetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx\_8u\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.22 NppStatus nppiMaxIndxGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiMaxIndx\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.92.2.23 NppStatus nppiMaxIndxGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C3R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.92.2.24 NppStatus nppiMaxIndxGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_16u\_C4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.92.2.25 NppStatus nppiMaxIndxGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_AC4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.92.2.26 NppStatus nppiMaxIndxGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C1R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.27 NppStatus nppiMaxIndxGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C3R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.28 NppStatus nppiMaxIndxGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_32f\_C4R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.29 NppStatus nppiMaxIndxGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_AC4R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.30 NppStatus nppiMaxIndxGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.31 NppStatus nppiMaxIndxGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.92.2.32 NppStatus nppiMaxIndxGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiMaxIndx\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.93 MinMax

Primitives for computing both the minimal and the maximal values of an image.

### MinMax

The functions require the device scratch buffer.

- `NppStatus nppiMinMax_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pMin, `Npp8u` \*pMax, `Npp8u` \*pDeviceBuffer)  
*One-channel 8-bit unsigned image MinMax.*
- `NppStatus nppiMinMax_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` \*pMin, `Npp16u` \*pMax, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit unsigned image MinMax.*
- `NppStatus nppiMinMax_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` \*pMin, `Npp16s` \*pMax, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit signed image MinMax.*
- `NppStatus nppiMinMax_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` \*pMin, `Npp32f` \*pMax, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit floating point image MinMax.*
- `NppStatus nppiMinMax_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[3], `Npp8u` aMax[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image MinMax.*
- `NppStatus nppiMinMax_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` aMin[3], `Npp16u` aMax[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image MinMax.*
- `NppStatus nppiMinMax_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[3], `Npp16s` aMax[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 16-bit signed image MinMax.*
- `NppStatus nppiMinMax_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[3], `Npp32f` aMax[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 32-bit floating point image MinMax.*
- `NppStatus nppiMinMax_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[3], `Npp8u` aMax[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image MinMax ignoring alpha channel.*
- `NppStatus nppiMinMax_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` aMin[3], `Npp16u` aMax[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image MinMax ignoring alpha channel.*
- `NppStatus nppiMinMax_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[3], `Npp16s` aMax[3], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit signed image MinMax ignoring alpha channel.*

- `NppStatus nppiMinMax_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[3], `Npp32f` aMax[3], `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image MinMax ignoring alpha channel.*

- `NppStatus nppiMinMax_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` aMin[4], `Npp8u` aMax[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image MinMax.*

- `NppStatus nppiMinMax_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` aMin[4], `Npp16u` aMax[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit unsigned image MinMax.*

- `NppStatus nppiMinMax_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16s` aMin[4], `Npp16s` aMax[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit signed image MinMax.*

- `NppStatus nppiMinMax_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` aMin[4], `Npp32f` aMax[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image MinMax.*

## MinMaxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinMax primitives.

- `NppStatus nppiMinMaxGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_8u_C1R`.*

- `NppStatus nppiMinMaxGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_16u_C1R`.*

- `NppStatus nppiMinMaxGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_16s_C1R`.*

- `NppStatus nppiMinMaxGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_32f_C1R`.*

- `NppStatus nppiMinMaxGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_8u_C3R`.*

- `NppStatus nppiMinMaxGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_16u_C3R`.*

- `NppStatus nppiMinMaxGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_16s_C3R`.*

- `NppStatus nppiMinMaxGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMinMax_32f_C3R`.*

- [NppStatus nppiMinMaxGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_8u\\_AC4R](#).*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_16u\\_AC4R](#).*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_16s\\_AC4R](#).*
- [NppStatus nppiMinMaxGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_32f\\_AC4R](#).*
- [NppStatus nppiMinMaxGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_8u\\_C4R](#).*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_16u\\_C4R](#).*
- [NppStatus nppiMinMaxGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_16s\\_C4R](#).*
- [NppStatus nppiMinMaxGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiMinMax\\_32f\\_C4R](#).*

### 7.93.1 Detailed Description

Primitives for computing both the minimal and the maximal values of an image.

### 7.93.2 Function Documentation

#### 7.93.2.1 [NppStatus nppiMinMax\\_16s\\_AC4R](#) (const [Npp16s](#) \* *pSrc*, int *nSrcStep*, [NppiSize](#) *oSizeROI*, [Npp16s](#) *aMin*[3], [Npp16s](#) *aMax*[3], [Npp8u](#) \* *pDeviceBuffer*)

Four-channel 16-bit signed image MinMax ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16s\\_AC4R](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.2 NppStatus nppiMinMax\_16s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16s \* *pMin*, Npp16s \* *pMax*, Npp8u \* *pDeviceBuffer*)

One-channel 16-bit signed image MinMax.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMin* Pointer to the computed minimal result.

*pMax* Pointer to the computed maximal result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.3 NppStatus nppiMinMax\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16s *aMin*[3], Npp16s *aMax*[3], Npp8u \* *pDeviceBuffer*)

Three-channel 16-bit signed image MinMax.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16s\\_C3R](#) to determine the minium number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.4 NppStatus nppiMinMax\_16s\_C4R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16s *aMin*[4], Npp16s *aMax*[4], Npp8u \* *pDeviceBuffer*)

Four-channel 16-bit signed image MinMax.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Array that contains the minima.



*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16s\\_C4R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.93.2.5 NppStatus nppiMinMax\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u aMin[3], Npp16u aMax[3], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image MinMax ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.93.2.6 NppStatus nppiMinMax\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u \* pMin, Npp16u \* pMax, Npp8u \* pDeviceBuffer)**

One-channel 16-bit unsigned image MinMax.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMin* Pointer to the computed minimal result.

*pMax* Pointer to the computed maximal result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.7 **NppStatus nppiMinMax\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16u *aMin*[3], Npp16u *aMax*[3], Npp8u \* *pDeviceBuffer*)

Three-channel 16-bit unsigned image MinMax.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.8 **NppStatus nppiMinMax\_16u\_C4R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp16u *aMin*[4], Npp16u *aMax*[4], Npp8u \* *pDeviceBuffer*)

Four-channel 16-bit unsigned image MinMax.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.9 **NppStatus nppiMinMax\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32f *aMin*[3], Npp32f *aMax*[3], Npp8u \* *pDeviceBuffer*)

Four-channel 32-bit floating point image MinMax ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.93.2.10 NppStatus nppiMinMax\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f \* pMin, Npp32f \* pMax, Npp8u \* pDeviceBuffer)**

One-channel 32-bit floating point image MinMax.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMin* Pointer to the computed minimal result.

*pMax* Pointer to the computed maximal result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.93.2.11 NppStatus nppiMinMax\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[3], Npp32f aMax[3], Npp8u \* pDeviceBuffer)**

Three-channel 32-bit floating point image MinMax.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.12 `NppStatus nppiMinMax_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f aMin[4], Npp32f aMax[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image MinMax.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_32f\\_C4R](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.13 `NppStatus nppiMinMax_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MinMax ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_8u\\_AC4R](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.93.2.14 `NppStatus nppiMinMax_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pMin, Npp8u * pMax, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image MinMax.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pMin* Pointer to the computed minimal result.

*pMax* Pointer to the computed maximal result.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.93.2.15 NppStatus nppiMinMax\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[3], Npp8u aMax[3], Npp8u \* pDeviceBuffer)**

Three-channel 8-bit unsigned image MinMax.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_8u\\_C3R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.93.2.16 NppStatus nppiMinMax\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u aMin[4], Npp8u aMax[4], Npp8u \* pDeviceBuffer)**

Four-channel 8-bit unsigned image MinMax.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aMin* Array that contains the minima.

*aMax* Array that contains the maxima.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxGetBufferHostSize\\_8u\\_C4R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.93.2.17 NppStatus nppiMinMaxGetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16s\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.18 NppStatus nppiMinMaxGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.19 NppStatus nppiMinMaxGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.20 NppStatus nppiMinMaxGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.21 NppStatus nppiMinMaxGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16u\\_AC4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.22 NppStatus nppiMinMaxGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16u\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.23 NppStatus nppiMinMaxGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16u\\_C3R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.24 NppStatus nppiMinMaxGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.25 NppStatus nppiMinMaxGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.26 NppStatus nppiMinMaxGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.27 NppStatus nppiMinMaxGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.28 NppStatus nppiMinMaxGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.29 NppStatus nppiMinMaxGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.30 NppStatus nppiMinMaxGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMax\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.31 NppStatus nppiMinMaxGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiMinMax\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.93.2.32 NppStatus nppiMinMaxGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiMinMax\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.94 MinMaxIndx

Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.

### MinMaxIndx

If there are several minima and maxima in the selected region of interest, the function returns ones on the top leftmost position.

The scratch buffer is required by the functions.

- `NppStatus nppiMinMaxIndx_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pMinValue, `Npp8u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit unsigned char image.*

- `NppStatus nppiMinMaxIndx_8s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8s` \*pMinValue, `Npp8s` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit signed char image.*

- `NppStatus nppiMinMaxIndx_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp16u` \*pMinValue, `Npp16u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 16-bit unsigned short image.*

- `NppStatus nppiMinMaxIndx_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32f` \*pMinValue, `Npp32f` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 32-bit floating point image.*

### Masked MinMaxIndx

See [Masked Operation](#).

- `NppStatus nppiMinMaxIndx_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pMinValue, `Npp8u` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*Masked one-channel 8-bit unsigned image MinMaxIndx.*

- `NppStatus nppiMinMaxIndx_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8s` \*pMinValue, `Npp8s` \*pMaxValue, `NppiPoint` \*pMinIndex, `NppiPoint` \*pMaxIndex, `Npp8u` \*pDeviceBuffer)

*Masked one-channel 8-bit signed image MinMaxIndx.*

- [NppStatus nppiMinMaxIdx\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp16u](#) \*pMinValue, [Npp16u](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Masked one-channel 16-bit unsigned image MinMaxIdx.*

- [NppStatus nppiMinMaxIdx\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp32f](#) \*pMinValue, [Npp32f](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Masked one-channel 32-bit floating point image MinMaxIdx.*

## Channel MinMaxIdx

See [Channel-of-Interest API](#).

- [NppStatus nppiMinMaxIdx\\_8u\\_C3CR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) \*pMinValue, [Npp8u](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Three-channel 8-bit unsigned image MinMaxIdx affecting only single channel.*

- [NppStatus nppiMinMaxIdx\\_8s\\_C3CR](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8s](#) \*pMinValue, [Npp8s](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Three-channel 8-bit signed image MinMaxIdx affecting only single channel.*

- [NppStatus nppiMinMaxIdx\\_16u\\_C3CR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp16u](#) \*pMinValue, [Npp16u](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Three-channel 16-bit unsigned image MinMaxIdx affecting only single channel.*

- [NppStatus nppiMinMaxIdx\\_32f\\_C3CR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, int nCOI, [Npp32f](#) \*pMinValue, [Npp32f](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Three-channel 32-bit floating point image MinMaxIdx affecting only single channel.*

## Masked Channel MinMaxIdx

See [Masked Operation](#) and [Channel-of-Interest API](#).

- [NppStatus nppiMinMaxIdx\\_8u\\_C3CMR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8u](#) \*pMinValue, [Npp8u](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image MinMaxIdx affecting only single channel.*

- [NppStatus nppiMinMaxIdx\\_8s\\_C3CMR](#) (const [Npp8s](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp8s](#) \*pMinValue, [Npp8s](#) \*pMaxValue, [NppiPoint](#) \*pMinIndex, [NppiPoint](#) \*pMaxIndex, [Npp8u](#) \*pDeviceBuffer)

*Masked three-channel 8-bit signed image MinMaxIdx affecting only single channel.*

- **NppStatus** **nppiMinMaxIndx\_16u\_C3CMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp16u** \*pMinValue, **Npp16u** \*pMaxValue, **NppiPoint** \*pMinIndex, **NppiPoint** \*pMaxIndex, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.*

- **NppStatus** **nppiMinMaxIndx\_32f\_C3CMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp32f** \*pMinValue, **Npp32f** \*pMaxValue, **NppiPoint** \*pMinIndex, **NppiPoint** \*pMaxIndex, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image MinMaxIndx affecting only single channel.*

## MinMaxIndxGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the MinMaxIndx primitives.

- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_8u\_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_8s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_8s\_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_16u\_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_32f\_C1R**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_8u\_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_8s\_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_16u\_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_32f\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_32f\_C1MR**.*
- **NppStatus** **nppiMinMaxIndxGetBufferHostSize\_8u\_C3CR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiMinMaxIndx\_8u\_C3CR**.*

- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_8s\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMinMaxIdx\\_8s\\_C3CR](#).*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_16u\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMinMaxIdx\\_16u\\_C3CR](#).*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_32f\\_C3CR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMinMaxIdx\\_32f\\_C3CR](#).*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMinMaxIdx\\_8u\\_C3CMR](#).*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMinMaxIdx\\_8s\\_C3CMR](#).*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMinMaxIdx\\_16u\\_C3CMR](#).*
- **NppStatus** [nppiMinMaxIdxGetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*[hpBufferSize](#))  
*Buffer size for [nppiMinMaxIdx\\_32f\\_C3CMR](#).*

### 7.94.1 Detailed Description

Primitives for computing the minimal and the maximal values with their indices (X and Y coordinates) of an image.

### 7.94.2 Function Documentation

**7.94.2.1** **NppStatus** [nppiMinMaxIdx\\_16u\\_C1MR](#) ([const](#) [Npp16u](#) \* [pSrc](#), [int](#) [nSrcStep](#), [const](#) [Npp8u](#) \* [pMask](#), [int](#) [nMaskStep](#), [NppiSize](#) [oSizeROI](#), [Npp16u](#) \* [pMinValue](#), [Npp16u](#) \* [pMaxValue](#), [NppiPoint](#) \* [pMinIndex](#), [NppiPoint](#) \* [pMaxIndex](#), [Npp8u](#) \* [pDeviceBuffer](#))

Masked one-channel 16-bit unsigned image MinMaxIdx.

#### Parameters:

- [pSrc](#) [Source-Image Pointer](#).
- [nSrcStep](#) [Source-Image Line Step](#).
- [pMask](#) [Mask-Image Pointer](#).
- [nMaskStep](#) [Mask-Image Line Step](#).
- [oSizeROI](#) [Region-of-Interest \(ROI\)](#).

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_16u\\_C1MR](#) to determine the minium number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e.,  $pMinIndex = \{0, 0\}$ ,  $pMaxIndex = \{0, 0\}$ ,  $pMinValue = 0$ ,  $pMaxValue = 0$ . If any of  $pMinValue$ ,  $pMaxValue$ ,  $pMinIndex$ , or  $pMaxIndex$  is not needed, zero pointer must be passed correspondingly.

#### 7.94.2.2 NppStatus nppiMinMaxIndx\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp16u \* pMinValue, Npp16u \* pMaxValue, NppiPoint \* pMinIndex, NppiPoint \* pMaxIndex, Npp8u \* pDeviceBuffer)

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 16-bit unsigned short image.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of  $pMinValue$ ,  $pMaxValue$ ,  $pMinIndex$ , or  $pMaxIndex$  is not needed, zero pointer must be passed correspondingly.

#### 7.94.2.3 NppStatus nppiMinMaxIndx\_16u\_C3CMR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp16u \* pMinValue, Npp16u \* pMaxValue, NppiPoint \* pMinIndex, NppiPoint \* pMaxIndex, Npp8u \* pDeviceBuffer)

Masked three-channel 16-bit unsigned image MinMaxIndx affecting only single channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize\\_16u\\_C3CMR](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

#### 7.94.2.4 NppStatus nppiMinMaxIdx\_16u\_C3CR (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp16u \* pMinValue, Npp16u \* pMaxValue, NppiPoint \* pMinIndex, NppiPoint \* pMaxIndex, Npp8u \* pDeviceBuffer)

Three-channel 16-bit unsigned image MinMaxIdx affecting only single channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIdxGetBufferHostSize\\_16u\\_C3CR](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.



**7.94.2.5** `NppStatus nppiMinMaxIndx_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image MinMaxIndx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pMinValue* Pointer to the minimum value.  
*pMaxValue* Pointer to the maximum value.  
*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.  
*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_32f\\_C1MR](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., `pMinIndex = {0, 0}`, `pMaxIndex = {0, 0}`, `pMinValue = 0`, `pMaxValue = 0`. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.94.2.6** `NppStatus nppiMinMaxIndx_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 32-bit floating point image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pMinValue* Pointer to the minimum value.  
*pMaxValue* Pointer to the maximum value.  
*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.  
*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_32f\\_C1R](#) to determine the minimum number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified. If any of `pMinValue`, `pMaxValue`, `pMinIndex`, or `pMaxIndex` is not needed, zero pointer must be passed correspondingly.

**7.94.2.7** `NppStatus nppiMinMaxIndx_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pMinValue* Pointer to the minimum value.  
*pMaxValue* Pointer to the maximum value.  
*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.  
*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_32f\\_C3CMR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., pMinIndex = {0, 0}, pMaxIndex = {0, 0}, pMinValue = 0, pMaxValue = 0. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

**7.94.2.8** `NppStatus nppiMinMaxIndx_32f_C3CR (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp32f * pMinValue, Npp32f * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MinMaxIndx affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pMinValue* Pointer to the minimum value.  
*pMaxValue* Pointer to the maximum value.  
*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.  
*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_32f\\_C3CR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

**7.94.2.9** `NppStatus nppiMinMaxIndx_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image MinMaxIndx.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C1MR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., pMinIndex = {0, 0}, pMaxIndex = {0, 0}, pMinValue = 0, pMaxValue = 0. If any of pMinValue, pMaxValue, pMinIndex, or pMaxIndex is not needed, zero pointer must be passed correspondingly.

**7.94.2.10** `NppStatus nppiMinMaxIndx_8s_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit signed char image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C1R](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.94.2.11** `NppStatus nppiMinMaxIndx_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image MinMaxIndx affecting only single channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C3CMR](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.94.2.12** `NppStatus nppiMinMaxIndx_8s_C3CR (const Npp8s * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8s * pMinValue, Npp8s * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MinMaxIndx affecting only single channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8s\\_C3CR](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.94.2.13** `NppStatus nppiMinMaxIndx_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image MinMaxIndx.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pMinValue* Pointer to the minimum value.

*pMaxValue* Pointer to the maximum value.

*pMinIndex* Pointer to the indices (X and Y coordinates) of the minimum value.

*pMaxIndex* Pointer to the indices (X and Y coordinates) of the maximum value.

*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8u\\_C1MR](#) to determine the minimum number of bytes required.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.94.2.14** `NppStatus nppiMinMaxIndx_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Computes the minimal and the maximal pixel values with their X and Y coordinates of 1-channel 8-bit unsigned char image.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pMinValue* Pointer to the minimum value.  
*pMaxValue* Pointer to the maximum value.  
*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.  
*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#). If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.94.2.15** `NppStatus nppiMinMaxIndx_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nCOI* [Channel\\_of\\_Interest Number](#).  
*pMinValue* Pointer to the minimum value.  
*pMaxValue* Pointer to the maximum value.  
*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.  
*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8u\\_C3CMR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If the mask is filled with zeros, then all the returned values are zeros, i.e., *pMinIndex* = {0, 0}, *pMaxIndex* = {0, 0}, *pMinValue* = 0, *pMaxValue* = 0. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.94.2.16** `NppStatus nppiMinMaxIndx_8u_C3CR (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int nCOI, Npp8u * pMinValue, Npp8u * pMaxValue, NppiPoint * pMinIndex, NppiPoint * pMaxIndex, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MinMaxIndx affecting only single channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nCOI* [Channel\\_of\\_Interest Number](#).  
*pMinValue* Pointer to the minimum value.  
*pMaxValue* Pointer to the maximum value.  
*pMinIndex* Pointer to the indicies (X and Y coordinates) of the minimum value.  
*pMaxIndex* Pointer to the indicies (X and Y coordinates) of the maximum value.  
*pDeviceBuffer* Buffer to a scratch memory. Use [nppiMinMaxIndxGetBufferHostSize\\_8u\\_C3CR](#) to determine the minium number of bytes required.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified. If any of *pMinValue*, *pMaxValue*, *pMinIndex*, or *pMaxIndex* is not needed, zero pointer must be passed correspondingly.

**7.94.2.17** `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1MR (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMinMaxIndx\\_16u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.18** `NppStatus nppiMinMaxIndxGetBufferHostSize_16u_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMinMaxIndx\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.94.2.19 NppStatus nppiMinMaxIndxGetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMinMaxIndx\\_16u\\_C3CMR](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.94.2.20 NppStatus nppiMinMaxIndxGetBufferHostSize\_16u\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMinMaxIndx\\_16u\\_C3CR](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.94.2.21 NppStatus nppiMinMaxIndxGetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMinMaxIndx\\_32f\\_C1MR](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.94.2.22 NppStatus nppiMinMaxIndxGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMinMaxIndx\\_32f\\_C1R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).



**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.23 NppStatus nppiMinMaxIndxGetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMaxIndx\\_32f\\_C3CMR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.24 NppStatus nppiMinMaxIndxGetBufferHostSize\_32f\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMaxIndx\\_32f\\_C3CR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.25 NppStatus nppiMinMaxIndxGetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMaxIndx\\_8s\\_C1MR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.26 NppStatus nppiMinMaxIndxGetBufferHostSize\_8s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMinMaxIndx\\_8s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.27 NppStatus nppiMinMaxIndxGetBufferHostSize\_8s\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMinMaxIndx\\_8s\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.28 NppStatus nppiMinMaxIndxGetBufferHostSize\_8s\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMinMaxIndx\\_8s\\_C3CR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.29 NppStatus nppiMinMaxIndxGetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMinMaxIndx\\_8u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.30 NppStatus nppiMinMaxIndxGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMaxIndx\\_8u\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.31 NppStatus nppiMinMaxIndxGetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMaxIndx\\_8u\\_C3CMR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.94.2.32 NppStatus nppiMinMaxIndxGetBufferHostSize\_8u\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMinMaxIndx\\_8u\\_C3CR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

## 7.95 Mean

Primitives for computing the arithmetic mean of all the pixel values in an image.

### Mean

Given an image  $pSrc$  with width  $W$  and height  $H$ , the arithmetic mean will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

The mean functions require additional scratch buffer for computations.

- `NppStatus nppiMean_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*One-channel 8-bit unsigned image Mean.*
- `NppStatus nppiMean_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*One-channel 16-bit unsigned image Mean.*
- `NppStatus nppiMean_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*One-channel 16-bit signed image Mean.*
- `NppStatus nppiMean_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*One-channel 32-bit floating point image Mean.*
- `NppStatus nppiMean_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])  
*Three-channel 8-bit unsigned image Mean.*
- `NppStatus nppiMean_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])  
*Three-channel 16-bit unsigned image Mean.*
- `NppStatus nppiMean_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])  
*Three-channel 16-bit signed image Mean.*
- `NppStatus nppiMean_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[3])  
*Three-channel 32-bit floating point image Mean.*
- `NppStatus nppiMean_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` aMean[4])  
*Four-channel 8-bit unsigned image Mean.*

- [NppStatus](#) [nppiMean\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) aMean[4])  
*Four-channel 16-bit unsigned image Mean.*
- [NppStatus](#) [nppiMean\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) aMean[4])  
*Four-channel 16-bit signed image Mean.*
- [NppStatus](#) [nppiMean\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) aMean[4])  
*Four-channel 32-bit floating point image Mean.*
- [NppStatus](#) [nppiMean\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) aMean[3])  
*Four-channel 8-bit unsigned image Mean ignoring alpha channel.*
- [NppStatus](#) [nppiMean\\_16u\\_AC4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) aMean[3])  
*Four-channel 16-bit unsigned image Mean ignoring alpha channel.*
- [NppStatus](#) [nppiMean\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) aMean[3])  
*Four-channel 16-bit signed image Mean ignoring alpha channel.*
- [NppStatus](#) [nppiMean\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) aMean[3])  
*Four-channel 32-bit floating point image Mean ignoring alpha channel.*

## Masked Mean

See [Masked Operation](#).

- [NppStatus](#) [nppiMean\\_8u\\_C1MR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean)  
*Masked one-channel 8-bit unsigned image Mean.*
- [NppStatus](#) [nppiMean\\_8s\\_C1MR](#) (const [Npp8s](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean)  
*Masked one-channel 8-bit signed image Mean.*
- [NppStatus](#) [nppiMean\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean)  
*Masked one-channel 16-bit unsigned image Mean.*
- [NppStatus](#) [nppiMean\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp8u](#) \*pDeviceBuffer, [Npp64f](#) \*pMean)  
*Masked one-channel 32-bit floating point image Mean.*

## Masked Channel Mean

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiMean_8u_C3CMR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*Masked three-channel 8-bit unsigned image Mean affecting only single channel.*
- `NppStatus nppiMean_8s_C3CMR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*Masked three-channel 8-bit signed image Mean affecting only single channel.*
- `NppStatus nppiMean_16u_C3CMR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*Masked three-channel 16-bit unsigned image Mean affecting only single channel.*
- `NppStatus nppiMean_32f_C3CMR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean)  
*Masked three-channel 32-bit floating point image Mean affecting only single channel.*

## MeanGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean primitives.

- `NppStatus nppiMeanGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_8u_C1R`.*
- `NppStatus nppiMeanGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16u_C1R`.*
- `NppStatus nppiMeanGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16s_C1R`.*
- `NppStatus nppiMeanGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_32f_C1R`.*
- `NppStatus nppiMeanGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_8u_C3R`.*
- `NppStatus nppiMeanGetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16u_C3R`.*
- `NppStatus nppiMeanGetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16s_C3R`.*
- `NppStatus nppiMeanGetBufferHostSize_32f_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_32f_C3R`.*
- `NppStatus nppiMeanGetBufferHostSize_8u_AC4R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiMean_8u_AC4R`.*

- `NppStatus nppiMeanGetBufferHostSize_16u_AC4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16u_AC4R`.*
- `NppStatus nppiMeanGetBufferHostSize_16s_AC4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16s_AC4R`.*
- `NppStatus nppiMeanGetBufferHostSize_32f_AC4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_32f_AC4R`.*
- `NppStatus nppiMeanGetBufferHostSize_8u_C4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_8u_C4R`.*
- `NppStatus nppiMeanGetBufferHostSize_16u_C4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16u_C4R`.*
- `NppStatus nppiMeanGetBufferHostSize_16s_C4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16s_C4R`.*
- `NppStatus nppiMeanGetBufferHostSize_32f_C4R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_32f_C4R`.*
- `NppStatus nppiMeanGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_8u_C1MR`.*
- `NppStatus nppiMeanGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_8s_C1MR`.*
- `NppStatus nppiMeanGetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16u_C1MR`.*
- `NppStatus nppiMeanGetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_32f_C1MR`.*
- `NppStatus nppiMeanGetBufferHostSize_8u_C3CMR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_8u_C3CMR`.*
- `NppStatus nppiMeanGetBufferHostSize_8s_C3CMR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_8s_C3CMR`.*
- `NppStatus nppiMeanGetBufferHostSize_16u_C3CMR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_16u_C3CMR`.*
- `NppStatus nppiMeanGetBufferHostSize_32f_C3CMR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Buffer size for `nppiMean_32f_C3CMR`.*

### 7.95.1 Detailed Description

Primitives for computing the arithmetic mean of all the pixel values in an image.

## 7.95.2 Function Documentation

### 7.95.2.1 NppStatus nppiMean\_16s\_AC4R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aMean*[3])

Four-channel 16-bit signed image Mean ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16s\\_AC4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.95.2.2 NppStatus nppiMean\_16s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*)

One-channel 16-bit signed image Mean.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16s\\_C1R](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.95.2.3 NppStatus nppiMean\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aMean*[3])

Three-channel 16-bit signed image Mean.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16s\\_C3R](#) to determine the minium number of bytes required.



*aMean* Array that contains the computed mean results.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.95.2.4 NppStatus nppiMean\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[4])**

Four-channel 16-bit signed image Mean.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16s\\_C4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.95.2.5 NppStatus nppiMean\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[3])**

Four-channel 16-bit unsigned image Mean ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_AC4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.95.2.6 NppStatus nppiMean\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)**

Masked one-channel 16-bit unsigned image Mean.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.95.2.7** `NppStatus nppiMean_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 16-bit unsigned image Mean.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.95.2.8** `NppStatus nppiMean_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 16-bit unsigned image Mean affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

### 7.95.2.9 NppStatus nppiMean\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aMean*[3])

Three-channel 16-bit unsigned image Mean.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C3R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.95.2.10 NppStatus nppiMean\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aMean*[4])

Four-channel 16-bit unsigned image Mean.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_16u\\_C4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.95.2.11 NppStatus nppiMean\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f *aMean*[3])

Four-channel 32-bit floating point image Mean ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_AC4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.95.2.12** `NppStatus nppiMean_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked one-channel 32-bit floating point image Mean.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.95.2.13** `NppStatus nppiMean_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 32-bit floating point image Mean.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.95.2.14** `NppStatus nppiMean_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 32-bit floating point image Mean affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.95.2.15** `NppStatus nppiMean_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Three-channel 32-bit floating point image Mean.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C3R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.95.2.16** `NppStatus nppiMean_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

Four-channel 32-bit floating point image Mean.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_32f\\_C4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.95.2.17 NppStatus nppiMean\_8s\_C1MR (const Npp8s \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)**

Masked one-channel 8-bit signed image Mean.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8s\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.95.2.18 NppStatus nppiMean\_8s\_C3CMR (const Npp8s \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)**

Masked three-channel 8-bit signed image Mean affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8s\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

#### 7.95.2.19 NppStatus nppiMean\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f aMean[3])

Four-channel 8-bit unsigned image Mean ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8u\\_AC4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.95.2.20 NppStatus nppiMean\_8u\_C1MR (const Npp8u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp8u \* pDeviceBuffer, Npp64f \* pMean)

Masked one-channel 8-bit unsigned image Mean.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8u\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.95.2.21 `NppStatus nppiMean_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

One-channel 8-bit unsigned image Mean.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMeanGetBufferHostSize_8u_C1R` to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.95.2.22 `NppStatus nppiMean_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp8u * pDeviceBuffer, Npp64f * pMean)`

Masked three-channel 8-bit unsigned image Mean affecting only single channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use `nppiMeanGetBufferHostSize_8u_C3CMR` to determine the minium number of bytes required.

*pMean* Pointer to the computed mean result.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

### 7.95.2.23 `NppStatus nppiMean_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[3])`

Three-channel 8-bit unsigned image Mean.

#### Parameters:

*pSrc* Source-Image Pointer.



*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8u\\_C3R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.95.2.24** `NppStatus nppiMean_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp8u * pDeviceBuffer, Npp64f aMean[4])`

Four-channel 8-bit unsigned image Mean.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanGetBufferHostSize\\_8u\\_C4R](#) to determine the minium number of bytes required.

*aMean* Array that contains the computed mean results.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.95.2.25** `NppStatus nppiMeanGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMean\\_16s\\_AC4R](#).

#### Parameters:

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.26** `NppStatus nppiMeanGetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiMean\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.27 NppStatus nppiMeanGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.28 NppStatus nppiMeanGetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.29 NppStatus nppiMeanGetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_16u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.30 NppStatus nppiMeanGetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_16u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.31 NppStatus nppiMeanGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.32 NppStatus nppiMeanGetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_16u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.33 NppStatus nppiMeanGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.34 NppStatus nppiMeanGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.35 NppStatus nppiMeanGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.36 NppStatus nppiMeanGetBufferHostSize\_32f\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_32f\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.37 NppStatus nppiMeanGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.38 NppStatus nppiMeanGetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_32f\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.39 NppStatus nppiMeanGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.40 NppStatus nppiMeanGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.41 NppStatus nppiMeanGetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_8s\\_C1MR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.42 NppStatus nppiMeanGetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_8s\\_C3CMR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.43 NppStatus nppiMeanGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_8u\\_AC4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.44 NppStatus nppiMeanGetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_8u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.45 NppStatus nppiMeanGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.46 NppStatus nppiMeanGetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_8u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.47 NppStatus nppiMeanGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.95.2.48 NppStatus nppiMeanGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiMean\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



## 7.96 Mean\_StdDev

Primitives for computing both the arithmetic mean and the standard deviation of an image.

### Mean\_StdDev

Given an image  $pSrc$  with width  $W$  and height  $H$ , the mean and the standard deviation will be computed as

$$Mean = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} pSrc(j, i)$$

$$StdDev = \sqrt{\frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} (pSrc(j, i) - Mean)^2}$$

The Mean\_StdDev primitives require additional scratch buffer for computations.

- `NppStatus nppiMean_StdDev_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)  
*One-channel 8-bit unsigned image Mean\_StdDev.*
- `NppStatus nppiMean_StdDev_8s_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)  
*One-channel 8-bit signed image Mean\_StdDev.*
- `NppStatus nppiMean_StdDev_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)  
*One-channel 16-bit unsigned image Mean\_StdDev.*
- `NppStatus nppiMean_StdDev_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)  
*One-channel 32-bit floating point image Mean\_StdDev.*

### Masked Mean\_StdDev

See [Masked Operation](#).

- `NppStatus nppiMean_StdDev_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)  
*Masked one-channel 8-bit unsigned image Mean\_StdDev.*
- `NppStatus nppiMean_StdDev_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)  
*Masked one-channel 8-bit signed image Mean\_StdDev.*
- `NppStatus nppiMean_StdDev_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)  
*Masked one-channel 16-bit unsigned image Mean\_StdDev.*

- `NppStatus nppiMean_StdDev_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Masked one-channel 32-bit floating point image Mean\_StdDev.*

## Channel Mean\_StdDev

See [Channel-of-Interest API](#).

- `NppStatus nppiMean_StdDev_8u_C3CR` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Three-channel 8-bit unsigned image Mean\_StdDev affecting only single channel.*

- `NppStatus nppiMean_StdDev_8s_C3CR` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Three-channel 8-bit signed image Mean\_StdDev affecting only single channel.*

- `NppStatus nppiMean_StdDev_16u_C3CR` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Three-channel 16-bit unsigned image Mean\_StdDev affecting only single channel.*

- `NppStatus nppiMean_StdDev_32f_C3CR` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Three-channel 32-bit floating point image Mean\_StdDev affecting only single channel.*

## Masked Channel Mean\_StdDev

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiMean_StdDev_8u_C3CMR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Masked three-channel 8-bit unsigned image Mean\_StdDev.*

- `NppStatus nppiMean_StdDev_8s_C3CMR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Masked three-channel 8-bit signed image Mean\_StdDev.*

- `NppStatus nppiMean_StdDev_16u_C3CMR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Masked three-channel 16-bit unsigned image Mean\_StdDev.*

- `NppStatus nppiMean_StdDev_32f_C3CMR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp8u` \*pDeviceBuffer, `Npp64f` \*pMean, `Npp64f` \*pStdDev)

*Masked three-channel 32-bit floating point image Mean\_StdDev.*

## MeanStdDevGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean\_StdDev primitives.

- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_8u\_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_8s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_8s\_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_16u\_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_32f\_C1R*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_8u\_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_8s\_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_16u\_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_32f\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_32f\_C1MR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_8u\_C3CR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_8u\_C3CR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_8s\_C3CR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_8s\_C3CR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_16u\_C3CR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_16u\_C3CR*.
- **NppStatus** **nppiMeanStdDevGetBufferHostSize\_32f\_C3CR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
Buffer size for *nppiMean\_StdDev\_32f\_C3CR*.

- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiMean\\_StdDev\\_8u\\_C3CMR](#).*
- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiMean\\_StdDev\\_8s\\_C3CMR](#).*
- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiMean\\_StdDev\\_16u\\_C3CMR](#).*
- [NppStatus](#) [nppiMeanStdDevGetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiMean\\_StdDev\\_32f\\_C3CMR](#).*

## 7.96.1 Detailed Description

Primitives for computing both the arithmetic mean and the standard deviation of an image.

## 7.96.2 Function Documentation

**7.96.2.1** [NppStatus](#) [nppiMean\\_StdDev\\_16u\\_C1MR](#) ([const](#) [Npp16u](#) \* [pSrc](#), [int](#) [nSrcStep](#), [const](#) [Npp8u](#) \* [pMask](#), [int](#) [nMaskStep](#), [NppiSize](#) [oSizeROI](#), [Npp8u](#) \* [pDeviceBuffer](#), [Npp64f](#) \* [pMean](#), [Npp64f](#) \* [pStdDev](#))

Masked one-channel 16-bit unsigned image Mean\_StdDev.

### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[pMask](#) [Mask-Image Pointer](#).

[nMaskStep](#) [Mask-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C1MR](#) to determine the minium number of bytes required.

[pMean](#) Pointer to the computed mean.

[pStdDev](#) Pointer to the computed standard deviation.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.96.2.2 NppStatus nppiMean\_StdDev\_16u\_C1R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

One-channel 16-bit unsigned image Mean\_StdDev.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C1R](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.96.2.3 NppStatus nppiMean\_StdDev\_16u\_C3CMR (const Npp16u \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Masked three-channel 16-bit unsigned image Mean\_StdDev.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

#### 7.96.2.4 NppStatus nppiMean\_StdDev\_16u\_C3CR (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Three-channel 16-bit unsigned image Mean\_StdDev affecting only single channel.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_16u\\_C3CR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

#### 7.96.2.5 NppStatus nppiMean\_StdDev\_32f\_C1MR (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Masked one-channel 32-bit floating point image Mean\_StdDev.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

### 7.96.2.6 NppStatus nppiMean\_StdDev\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

One-channel 32-bit floating point image Mean\_StdDev.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C1R](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

### 7.96.2.7 NppStatus nppiMean\_StdDev\_32f\_C3CMR (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Masked three-channel 32-bit floating point image Mean\_StdDev.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#) Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

### 7.96.2.8 **NppStatus nppiMean\_StdDev\_32f\_C3CR** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Three-channel 32-bit floating point image Mean\_StdDev affecting only single channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_32f\\_C3CR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

### 7.96.2.9 **NppStatus nppiMean\_StdDev\_8s\_C1MR** (const Npp8s \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Masked one-channel 8-bit signed image Mean\_StdDev.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



### 7.96.2.10 NppStatus nppiMean\_StdDev\_8s\_C1R (const Npp8s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

One-channel 8-bit signed image Mean\_StdDev.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C1R](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.96.2.11 NppStatus nppiMean\_StdDev\_8s\_C3CMR (const Npp8s \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Masked three-channel 8-bit signed image Mean\_StdDev.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

### 7.96.2.12 NppStatus nppiMean\_StdDev\_8s\_C3CR (const Npp8s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Three-channel 8-bit signed image Mean\_StdDev affecting only single channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8s\\_C3CR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

### 7.96.2.13 NppStatus nppiMean\_StdDev\_8u\_C1MR (const Npp8u \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Masked one-channel 8-bit unsigned image Mean\_StdDev.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8u\\_C1MR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.96.2.14 NppStatus nppiMean\_StdDev\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

One-channel 8-bit unsigned image Mean\_StdDev.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8u\\_C1R](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.96.2.15 NppStatus nppiMean\_StdDev\_8u\_C3CMR (const Npp8u \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Masked three-channel 8-bit unsigned image Mean\_StdDev.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8u\\_C3CMR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

### 7.96.2.16 NppStatus nppiMean\_StdDev\_8u\_C3CR (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, int *nCOI*, Npp8u \* *pDeviceBuffer*, Npp64f \* *pMean*, Npp64f \* *pStdDev*)

Three-channel 8-bit unsigned image Mean\_StdDev affecting only single channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#)  
Use [nppiMeanStdDevGetBufferHostSize\\_8u\\_C3CR](#) to determine the minium number of bytes required.

*pMean* Pointer to the computed mean.

*pStdDev* Pointer to the computed standard deviation.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

### 7.96.2.17 NppStatus nppiMeanStdDevGetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMean\\_StdDev\\_16u\\_C1MR](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.96.2.18 NppStatus nppiMeanStdDevGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiMean\\_StdDev\\_16u\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.19 NppStatus nppiMeanStdDevGetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_StdDev\\_16u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.20 NppStatus nppiMeanStdDevGetBufferHostSize\_16u\_C3CR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_StdDev\\_16u\\_C3CR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.21 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_StdDev\\_32f\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.22 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiMean\\_StdDev\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.23 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_32f\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.24 NppStatus nppiMeanStdDevGetBufferHostSize\_32f\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_32f\\_C3CR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.25 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8s\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.26 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.27 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8s\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.28 NppStatus nppiMeanStdDevGetBufferHostSize\_8s\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8s\\_C3CR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.29 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.30 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.31 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.96.2.32 NppStatus nppiMeanStdDevGetBufferHostSize\_8u\_C3CR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiMean\\_StdDev\\_8u\\_C3CR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



## 7.97 Image Norms

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.

### Modules

- [Norm\\_Inf](#)

*Primitives for computing the infinity norm of an image.*

- [Norm\\_L1](#)

*Primitives for computing the L1 norm of an image.*

- [Norm\\_L2](#)

*Primitives for computing the L2 norm of an image.*

- [NormDiff\\_Inf](#)

*Primitives for computing the infinity norm of difference of pixels between two images.*

- [NormDiff\\_L1](#)

*Primitives for computing the L1 norm of difference of pixels between two images.*

- [NormDiff\\_L2](#)

*Primitives for computing the L2 norm of difference of pixels between two images.*

- [NormRel\\_Inf](#)

*Primitives for computing the relative error of infinity norm between two images.*

- [NormRel\\_L1](#)

*Primitives for computing the relative error of L1 norm between two images.*

- [NormRel\\_L2](#)

*Primitives for computing the relative error of L2 norm between two images.*

### 7.97.1 Detailed Description

Primitives for computing the norms of an image, the norms of difference, and the relative errors of two images.

Given an image  $pSrc$  with width  $W$  and height  $H$ ,

1. The infinity norm ([Norm\\_Inf](#)) is defined as the largest absolute pixel value of the image.
2. The L1 norm ([Norm\\_L1](#)) is defined as the sum of the absolute pixel value of the image, i.e.,

$$Norm\_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|$$

3. The L2 norm (Norm\_L2) is defined as the square root of the sum of the squared absolute pixel value of the image, i.e.,

$$Norm\_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc(j, i)|^2}$$

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*,

1. The infinity norm of difference (NormDiff\_Inf) is defined as the largest absolute difference between pixels of two images.
2. The L1 norm of difference (NormDiff\_L1) is defined as the sum of the absolute difference between pixels of two images, i.e.,

$$NormDiff\_L1 = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

3. The L2 norm of difference (NormDiff\_L2) is defined as the squared root of the sum of the squared absolute difference between pixels of two images, i.e.,

$$NormDiff\_L2 = \sqrt{\sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|^2}$$

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*,

1. The relative error for the infinity norm of difference (NormRel\_Inf) is defined as NormDiff\_Inf divided by the infinity norm of the second image, i.e.,

$$NormRel\_Inf = \frac{NormDiff\_Inf}{Norm\_Inf_{src2}}$$

2. The relative error for the L1 norm of difference (NormRel\_L1) is defined as NormDiff\_L1 divided by the L1 norm of the second image, i.e.,

$$NormRel\_L1 = \frac{NormDiff\_L1}{Norm\_L1_{src2}}$$

3. The relative error for the L2 norm of difference (NormRel\_L2) is defined as NormDiff\_L2 divided by the L2 norm of the second image, i.e.,

$$NormRel\_L2 = \frac{NormDiff\_L2}{Norm\_L2_{src2}}$$

The norm functions require the addition device scratch buffer for the computations.

## 7.98 Norm\_Inf

Primitives for computing the infinity norm of an image.

### Basic Norm\_Inf

- **NppStatus nppiNorm\_Inf\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_32s\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image Norm\_Inf.*
- **NppStatus nppiNorm\_Inf\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image Norm\_Inf ignoring alpha channel.*
- **NppStatus nppiNorm\_Inf\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp64f** aNorm[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image Norm\_Inf ignoring alpha channel.*

- `NppStatus nppiNorm_Inf_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit signed image Norm\_Inf ignoring alpha channel.*
- `NppStatus nppiNorm_Inf_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 32-bit floating point image Norm\_Inf ignoring alpha channel.*
- `NppStatus nppiNorm_Inf_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image Norm\_Inf.*
- `NppStatus nppiNorm_Inf_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image Norm\_Inf.*
- `NppStatus nppiNorm_Inf_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit signed image Norm\_Inf.*
- `NppStatus nppiNorm_Inf_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 32-bit floating point image Norm\_Inf.*

## Masked Norm\_Inf

See [Masked Operation](#).

- `NppStatus nppiNorm_Inf_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 8-bit unsigned image Norm\_Inf.*
- `NppStatus nppiNorm_Inf_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 8-bit signed image Norm\_Inf.*
- `NppStatus nppiNorm_Inf_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 16-bit unsigned image Norm\_Inf.*
- `NppStatus nppiNorm_Inf_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 32-bit floating point image Norm\_Inf.*

## Masked Channel Norm\_Inf

See [Channel-of-Interest API](#) and [Masked Operation](#).

- **NppStatus nppiNorm\_Inf\_8u\_C3CMR** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*Masked three-channel 8-bit unsigned image Norm\_Inf affecting only single channel.*
- **NppStatus nppiNorm\_Inf\_8s\_C3CMR** (const **Npp8s** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*Masked three-channel 8-bit signed image Norm\_Inf affecting only single channel.*
- **NppStatus nppiNorm\_Inf\_16u\_C3CMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*Masked three-channel 16-bit unsigned image Norm\_Inf affecting only single channel.*
- **NppStatus nppiNorm\_Inf\_32f\_C3CMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*Masked three-channel 32-bit floating point image Norm\_Inf affecting only single channel.*

## NormInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm\_Inf primitives.

- **NppStatus nppiNormInfGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_8u\_C1R.*
- **NppStatus nppiNormInfGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_16u\_C1R.*
- **NppStatus nppiNormInfGetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_16s\_C1R.*
- **NppStatus nppiNormInfGetBufferHostSize\_32s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_32s\_C1R.*
- **NppStatus nppiNormInfGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_32f\_C1R.*
- **NppStatus nppiNormInfGetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_8u\_C1MR.*
- **NppStatus nppiNormInfGetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_8s\_C1MR.*
- **NppStatus nppiNormInfGetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_16u\_C1MR.*
- **NppStatus nppiNormInfGetBufferHostSize\_32f\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for nppiNorm\_Inf\_32f\_C1MR.*
- **NppStatus nppiNormInfGetBufferHostSize\_8u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_Inf_8u_C3R`.*

- `NppStatus nppiNormInfGetBufferHostSize_16u_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_16u_C3R`.*
- `NppStatus nppiNormInfGetBufferHostSize_16s_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_16s_C3R`.*
- `NppStatus nppiNormInfGetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_32f_C3R`.*
- `NppStatus nppiNormInfGetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_8u_AC4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_16u_AC4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_16s_AC4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_32f_AC4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_8u_C4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_16u_C4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_16s_C4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_32f_C4R`.*
- `NppStatus nppiNormInfGetBufferHostSize_8u_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_8u_C3CMR`.*
- `NppStatus nppiNormInfGetBufferHostSize_8s_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_8s_C3CMR`.*
- `NppStatus nppiNormInfGetBufferHostSize_16u_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_16u_C3CMR`.*
- `NppStatus nppiNormInfGetBufferHostSize_32f_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_Inf_32f_C3CMR`.*

### 7.98.1 Detailed Description

Primitives for computing the infinity norm of an image.

### 7.98.2 Function Documentation

#### 7.98.2.1 NppStatus nppiNorm\_Inf\_16s\_AC4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)

Four-channel 16-bit signed image Norm\_Inf ignoring alpha channel.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.98.2.2 NppStatus nppiNorm\_Inf\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)

One-channel 16-bit signed image Norm\_Inf.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.98.2.3 NppStatus nppiNorm\_Inf\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)

Three-channel 16-bit signed image Norm\_Inf.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.4 NppStatus nppiNorm\_Inf\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit signed image Norm\_Inf.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.5 NppStatus nppiNorm\_Inf\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image Norm\_Inf ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.98.2.6** `NppStatus nppiNorm_Inf_16u_C1MR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image Norm\_Inf.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.7** `NppStatus nppiNorm_Inf_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Norm\_Inf.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.8** `NppStatus nppiNorm_Inf_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image Norm\_Inf affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.98.2.9 NppStatus nppiNorm\_Inf\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Three-channel 16-bit unsigned image Norm\_Inf.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.10 NppStatus nppiNorm\_Inf\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image Norm\_Inf.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.98.2.11 NppStatus nppiNorm\_Inf\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u \* *pDeviceBuffer*)

Four-channel 32-bit floating point image Norm\_Inf ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.98.2.12 NppStatus nppiNorm\_Inf\_32f\_C1MR (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

Masked one-channel 32-bit floating point image Norm\_Inf.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.98.2.13 NppStatus nppiNorm\_Inf\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

One-channel 32-bit floating point image Norm\_Inf.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.14** `NppStatus nppiNorm_Inf_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image Norm\_Inf affecting only single channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.98.2.15** `NppStatus nppiNorm_Inf_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Norm\_Inf.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.98.2.16 NppStatus nppiNorm\_Inf\_32f\_C4R (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u \* *pDeviceBuffer*)

Four-channel 32-bit floating point image Norm\_Inf.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.98.2.17 NppStatus nppiNorm\_Inf\_32s\_C1R (const Npp32s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

One-channel 32-bit signed image Norm\_Inf.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_32s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.98.2.18 NppStatus nppiNorm\_Inf\_8s\_C1MR (const Npp8s \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

Masked one-channel 8-bit signed image Norm\_Inf.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.19** `NppStatus nppiNorm_Inf_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm\_Inf affecting only single channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.98.2.20** `NppStatus nppiNorm_Inf_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm\_Inf ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.21** `NppStatus nppiNorm_Inf_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image Norm\_Inf.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.22** `NppStatus nppiNorm_Inf_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Norm\_Inf.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.23** `NppStatus nppiNorm_Inf_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm\_Inf affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.98.2.24** `NppStatus nppiNorm_Inf_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Norm\_Inf.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.98.2.25** `NppStatus nppiNorm_Inf_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm\_Inf.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormInfGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.98.2.26 NppStatus nppiNormInfGetBufferHostSize\_16s\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16s\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.27 NppStatus nppiNormInfGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.28 NppStatus nppiNormInfGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.29 NppStatus nppiNormInfGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.30 NppStatus nppiNormInfGetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.31 NppStatus nppiNormInfGetBufferHostSize\_16u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.32 NppStatus nppiNormInfGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.33 NppStatus nppiNormInfGetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.34 NppStatus nppiNormInfGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.35 NppStatus nppiNormInfGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.36 NppStatus nppiNormInfGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.37 NppStatus nppiNormInfGetBufferHostSize\_32f\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_32f\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.38 NppStatus nppiNormInfGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.39 NppStatus nppiNormInfGetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_32f\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.40 NppStatus nppiNormInfGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.41 NppStatus nppiNormInfGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.42 NppStatus nppiNormInfGetBufferHostSize\_32s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_32s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.43 NppStatus nppiNormInfGetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_8s\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.44 NppStatus nppiNormInfGetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_8s\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.45 NppStatus nppiNormInfGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.46 NppStatus nppiNormInfGetBufferHostSize\_8u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_Inf\\_8u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.47 NppStatus nppiNormInfGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiNorm\\_Inf\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.48 NppStatus nppiNormInfGetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiNorm\\_Inf\\_8u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.49 NppStatus nppiNormInfGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiNorm\\_Inf\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.98.2.50 NppStatus nppiNormInfGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiNorm\\_Inf\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



## 7.99 Norm\_L1

Primitives for computing the L1 norm of an image.

### Basic Norm\_L1

- `NppStatus nppiNorm_L1_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 8-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit signed image Norm\_L1.*
- `NppStatus nppiNorm_L1_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit floating point image Norm\_L1.*
- `NppStatus nppiNorm_L1_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 16-bit signed image Norm\_L1.*
- `NppStatus nppiNorm_L1_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 32-bit floating point image Norm\_L1.*
- `NppStatus nppiNorm_L1_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image Norm\_L1 ignoring alpha channel.*
- `NppStatus nppiNorm_L1_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image Norm\_L1 ignoring alpha channel.*
- `NppStatus nppiNorm_L1_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit signed image Norm\_L1 ignoring alpha channel.*

- `NppStatus nppiNorm_L1_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 32-bit floating point image Norm\_L1 ignoring alpha channel.*
- `NppStatus nppiNorm_L1_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit signed image Norm\_L1.*
- `NppStatus nppiNorm_L1_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)  
*Four-channel 32-bit floating point image Norm\_L1.*

## Masked Norm\_L1

See [Masked Operation](#).

- `NppStatus nppiNorm_L1_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 8-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 8-bit signed image Norm\_L1.*
- `NppStatus nppiNorm_L1_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 16-bit unsigned image Norm\_L1.*
- `NppStatus nppiNorm_L1_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked one-channel 32-bit floating point image Norm\_L1.*

## Masked Channel Norm\_L1

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiNorm_L1_8u_C3CMR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*Masked three-channel 8-bit unsigned image Norm\_L1 affecting only single channel.*

- **NppStatus** **nppiNorm\_L1\_8s\_C3CMR** (const **Npp8s** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*Masked three-channel 8-bit signed image Norm\_L1 affecting only single channel.*
- **NppStatus** **nppiNorm\_L1\_16u\_C3CMR** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*Masked three-channel 16-bit unsigned image Norm\_L1 affecting only single channel.*
- **NppStatus** **nppiNorm\_L1\_32f\_C3CMR** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*Masked three-channel 32-bit floating point image Norm\_L1 affecting only single channel.*

## NormL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm\_L1 primitives.

- **NppStatus** **nppiNormL1GetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_8u\_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_16u\_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_16s\_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_32f\_C1R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_8u\_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_8s\_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_16u\_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_32f\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_32f\_C1MR**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_8u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_8u\_C3R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_16u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_16u\_C3R**.*
- **NppStatus** **nppiNormL1GetBufferHostSize\_16s\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNorm\_L1\_16s\_C3R**.*

- [NppStatus nppiNormL1GetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_32f\\_C3R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_8u\\_AC4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_16u\\_AC4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_16s\\_AC4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_32f\\_AC4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_8u\\_C4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_16u\\_C4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_16s\\_C4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_32f\\_C4R](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_8u\\_C3CMR](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_8s\\_C3CMR](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_16u\\_C3CMR](#).*
- [NppStatus nppiNormL1GetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Buffer size for [nppiNorm\\_L1\\_32f\\_C3CMR](#).*

### 7.99.1 Detailed Description

Primitives for computing the L1 norm of an image.

## 7.99.2 Function Documentation

### 7.99.2.1 NppStatus nppiNorm\_L1\_16s\_AC4R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u \* *pDeviceBuffer*)

Four-channel 16-bit signed image Norm\_L1 ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.99.2.2 NppStatus nppiNorm\_L1\_16s\_C1R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

One-channel 16-bit signed image Norm\_L1.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.99.2.3 NppStatus nppiNorm\_L1\_16s\_C3R (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u \* *pDeviceBuffer*)

Three-channel 16-bit signed image Norm\_L1.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.4 NppStatus nppiNorm\_L1\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit signed image Norm\_L1.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.5 NppStatus nppiNorm\_L1\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image Norm\_L1 ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.6 NppStatus nppiNorm\_L1\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

Masked one-channel 16-bit unsigned image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL1GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.7** `NppStatus nppiNorm_L1_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL1GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.8** `NppStatus nppiNorm_L1_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image Norm\_L1 affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.99.2.9 NppStatus nppiNorm\_L1\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Three-channel 16-bit unsigned image Norm\_L1.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.10 NppStatus nppiNorm\_L1\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image Norm\_L1.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.11 NppStatus nppiNorm\_L1\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Four-channel 32-bit floating point image Norm\_L1 ignoring alpha channel.



**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNorm* Array that contains the norm values of Three-channels.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL1GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.12** `NppStatus nppiNorm_L1_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL1GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.13** `NppStatus nppiNorm_L1_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL1GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.14** `NppStatus nppiNorm_L1_32f_C3CMR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image Norm\_L1 affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4, or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.99.2.15** `NppStatus nppiNorm_L1_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.16** `NppStatus nppiNorm_L1_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.17** `NppStatus nppiNorm_L1_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.18** `NppStatus nppiNorm_L1_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm\_L1 affecting only single channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.99.2.19** `NppStatus nppiNorm_L1_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm\_L1 ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.20** `NppStatus nppiNorm_L1_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.21** `NppStatus nppiNorm_L1_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Norm\_L1.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.99.2.22** `NppStatus nppiNorm_L1_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm\_L1 affecting only single channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.99.2.23** `NppStatus nppiNorm_L1_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Norm\_L1.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.99.2.24 **NppStatus nppiNorm\_L1\_8u\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u \* *pDeviceBuffer*)

Four-channel 8-bit unsigned image Norm\_L1.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL1GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.99.2.25 **NppStatus nppiNormL1GetBufferHostSize\_16s\_AC4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L1\\_16s\\_AC4R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.99.2.26 **NppStatus nppiNormL1GetBufferHostSize\_16s\_C1R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L1\\_16s\\_C1R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.27 NppStatus nppiNormL1GetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.28 NppStatus nppiNormL1GetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.29 NppStatus nppiNormL1GetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_16u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.30 NppStatus nppiNormL1GetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_16u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.31 NppStatus nppiNormL1GetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.32 NppStatus nppiNormL1GetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_16u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.33 NppStatus nppiNormL1GetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.99.2.34 NppStatus nppiNormL1GetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.35 NppStatus nppiNormL1GetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.36 NppStatus nppiNormL1GetBufferHostSize\_32f\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_32f\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.37 NppStatus nppiNormL1GetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.38 NppStatus nppiNormL1GetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_32f\\_C3CMR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.39 NppStatus nppiNormL1GetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_32f\\_C3R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.40 NppStatus nppiNormL1GetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_32f\\_C4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.41 NppStatus nppiNormL1GetBufferHostSize\_8s\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_8s\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.42 NppStatus nppiNormL1GetBufferHostSize\_8s\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_8s\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.43 NppStatus nppiNormL1GetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.44 NppStatus nppiNormL1GetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L1\\_8u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.45 NppStatus nppiNormL1GetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.46 NppStatus nppiNormL1GetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_8u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.47 NppStatus nppiNormL1GetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L1\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.99.2.48** `NppStatus nppiNormL1GetBufferHostSize_8u_C4R (NppiSize oSizeROI, int *  
hpBufferSize)`

Buffer size for [nppiNorm\\_L1\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.100 Norm\_L2

Primitives for computing the L2 norm of an image.

### Basic Norm\_L2

Computes the L2 norm of an image.

- `NppStatus nppiNorm_L2_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 8-bit unsigned image Norm\_L2.*
- `NppStatus nppiNorm_L2_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit unsigned image Norm\_L2.*
- `NppStatus nppiNorm_L2_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit signed image Norm\_L2.*
- `NppStatus nppiNorm_L2_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit floating point image Norm\_L2.*
- `NppStatus nppiNorm_L2_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image Norm\_L2.*
- `NppStatus nppiNorm_L2_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image Norm\_L2.*
- `NppStatus nppiNorm_L2_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 16-bit signed image Norm\_L2.*
- `NppStatus nppiNorm_L2_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 32-bit floating point image Norm\_L2.*
- `NppStatus nppiNorm_L2_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image Norm\_L2 ignoring alpha channel.*
- `NppStatus nppiNorm_L2_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image Norm\_L2 ignoring alpha channel.*
- `NppStatus nppiNorm_L2_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit signed image Norm\_L2 ignoring alpha channel.*

- `NppStatus nppiNorm_L2_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[3], `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image Norm\_L2 ignoring alpha channel.*

- `NppStatus nppiNorm_L2_8u_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image Norm\_L2.*

- `NppStatus nppiNorm_L2_16u_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit unsigned image Norm\_L2.*

- `NppStatus nppiNorm_L2_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit signed image Norm\_L2.*

- `NppStatus nppiNorm_L2_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp64f` aNorm[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image Norm\_L2.*

## Masked Norm\_L2

See [Masked Operation](#).

- `NppStatus nppiNorm_L2_8u_C1MR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked one-channel 8-bit unsigned image Norm\_L2.*

- `NppStatus nppiNorm_L2_8s_C1MR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked one-channel 8-bit signed image Norm\_L2.*

- `NppStatus nppiNorm_L2_16u_C1MR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked one-channel 16-bit unsigned image Norm\_L2.*

- `NppStatus nppiNorm_L2_32f_C1MR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked one-channel 32-bit floating point image Norm\_L2.*

## Masked Channel Norm\_L2

See [Channel-of-Interest API](#) and [Masked Operation](#).

- `NppStatus nppiNorm_L2_8u_C3CMR` (const `Npp8u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image Norm\_L2.*

- `NppStatus nppiNorm_L2_8s_C3CMR` (const `Npp8s` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 8-bit signed image Norm\_L2.*

- `NppStatus nppiNorm_L2_16u_C3CMR` (const `Npp16u` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image Norm\_L2.*

- `NppStatus nppiNorm_L2_32f_C3CMR` (const `Npp32f` \*pSrc, int nSrcStep, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNorm, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image Norm\_L2.*

## NormL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Norm\_L2 primitives.

- `NppStatus nppiNormL2GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_8u_C1R`.*

- `NppStatus nppiNormL2GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_16u_C1R`.*

- `NppStatus nppiNormL2GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_16s_C1R`.*

- `NppStatus nppiNormL2GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_32f_C1R`.*

- `NppStatus nppiNormL2GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_8u_C1MR`.*

- `NppStatus nppiNormL2GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_8s_C1MR`.*

- `NppStatus nppiNormL2GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_16u_C1MR`.*

- `NppStatus nppiNormL2GetBufferHostSize_32f_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_32f_C1MR`.*

- `NppStatus nppiNormL2GetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_8u_C3R`.*

- `NppStatus nppiNormL2GetBufferHostSize_16u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Buffer size for `nppiNorm_L2_16u_C3R`.*

- `NppStatus nppiNormL2GetBufferHostSize_16s_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)



*Buffer size for `nppiNorm_L2_16s_C3R`.*

- `NppStatus nppiNormL2GetBufferHostSize_32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_32f_C3R`.*
- `NppStatus nppiNormL2GetBufferHostSize_8u_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_8u_AC4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_16u_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_16u_AC4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_16s_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_16s_AC4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_32f_AC4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_8u_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_8u_C4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_16u_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_16u_C4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_16s_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_16s_C4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_32f_C4R`.*
- `NppStatus nppiNormL2GetBufferHostSize_8u_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_8u_C3CMR`.*
- `NppStatus nppiNormL2GetBufferHostSize_8s_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_8s_C3CMR`.*
- `NppStatus nppiNormL2GetBufferHostSize_16u_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_16u_C3CMR`.*
- `NppStatus nppiNormL2GetBufferHostSize_32f_C3CMR` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)  
*Buffer size for `nppiNorm_L2_32f_C3CMR`.*

### 7.100.1 Detailed Description

Primitives for computing the L2 norm of an image.

## 7.100.2 Function Documentation

### 7.100.2.1 `NppStatus nppiNorm_L2_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Norm\_L2 ignoring alpha channel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.100.2.2 `NppStatus nppiNorm_L2_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image Norm\_L2.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.100.2.3 `NppStatus nppiNorm_L2_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Norm\_L2.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.4 NppStatus nppiNorm\_L2\_16s\_C4R (const Npp16s \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit signed image Norm\_L2.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.5 NppStatus nppiNorm\_L2\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image Norm\_L2 ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.6 NppStatus nppiNorm\_L2\_16u\_C1MR (const Npp16u \* pSrc, int nSrcStep, const Npp8u \* pMask, int nMaskStep, NppiSize oSizeROI, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

Masked one-channel 16-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL2GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.7** `NppStatus nppiNorm_L2_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL2GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.8** `NppStatus nppiNorm_L2_16u_C3CMR (const Npp16u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.100.2.9 NppStatus nppiNorm\_L2\_16u\_C3R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Three-channel 16-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.10 NppStatus nppiNorm\_L2\_16u\_C4R (const Npp16u \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[4], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.11 NppStatus nppiNorm\_L2\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u \* pDeviceBuffer)**

Four-channel 32-bit floating point image Norm\_L2 ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNorm* Array that contains the norm values of Three-channels.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL2GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.12** `NppStatus nppiNorm_L2_32f_C1MR (const Npp32f * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image Norm\_L2.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL2GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if the step of the source image cannot be divided by 4.

**7.100.2.13** `NppStatus nppiNorm_L2_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Norm\_L2.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pNorm* Pointer to the norm value.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormL2GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.14** **NppStatus nppiNorm\_L2\_32f\_C3CMR** (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, int *nCOI*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

Masked three-channel 32-bit floating point image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if the step of the source image cannot be divided by 4, or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.100.2.15** **NppStatus nppiNorm\_L2\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[3], Npp8u \* *pDeviceBuffer*)

Three-channel 32-bit floating point image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.16** **NppStatus nppiNorm\_L2\_32f\_C4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u \* *pDeviceBuffer*)

Four-channel 32-bit floating point image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.17** `NppStatus nppiNorm_L2_8s_C1MR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.18** `NppStatus nppiNorm_L2_8s_C3CMR (const Npp8s * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.



**7.100.2.19** `NppStatus nppiNorm_L2_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Norm\_L2 ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.20** `NppStatus nppiNorm_L2_8u_C1MR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.21** `NppStatus nppiNorm_L2_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.100.2.22** `NppStatus nppiNorm_L2_8u_C3CMR (const Npp8u * pSrc, int nSrcStep, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNorm* Pointer to the norm value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.100.2.23** `NppStatus nppiNorm_L2_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp64f aNorm[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Norm\_L2.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Three-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.100.2.24 NppStatus nppiNorm\_L2\_8u\_C4R (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp64f *aNorm*[4], Npp8u \* *pDeviceBuffer*)

Four-channel 8-bit unsigned image Norm\_L2.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNorm* Array that contains the norm values of Four-channels.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormL2GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.100.2.25 NppStatus nppiNormL2GetBufferHostSize\_16s\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L2\\_16s\\_AC4R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.100.2.26 NppStatus nppiNormL2GetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L2\\_16s\\_C1R](#).

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.100.2.27 NppStatus nppiNormL2GetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L2\\_16s\\_C3R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.100.2.28 NppStatus nppiNormL2GetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L2\\_16s\\_C4R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.100.2.29 NppStatus nppiNormL2GetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L2\\_16u\\_AC4R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.100.2.30 NppStatus nppiNormL2GetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size for [nppiNorm\\_L2\\_16u\\_C1MR](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.31 NppStatus nppiNormL2GetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.32 NppStatus nppiNormL2GetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_16u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.33 NppStatus nppiNormL2GetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.34 NppStatus nppiNormL2GetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.35 NppStatus nppiNormL2GetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.36 NppStatus nppiNormL2GetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_32f\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.37 NppStatus nppiNormL2GetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.38 NppStatus nppiNormL2GetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_32f\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.39 NppStatus nppiNormL2GetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.40 NppStatus nppiNormL2GetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.41 NppStatus nppiNormL2GetBufferHostSize\_8s\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_8s\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.42 NppStatus nppiNormL2GetBufferHostSize\_8s\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_8s\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.43 NppStatus nppiNormL2GetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.44 NppStatus nppiNormL2GetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_8u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.45 NppStatus nppiNormL2GetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.46 NppStatus nppiNormL2GetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_8u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.47 NppStatus nppiNormL2GetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNorm\\_L2\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.100.2.48 NppStatus nppiNormL2GetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Buffer size for [nppiNorm\\_L2\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.101 NormDiff\_Inf

Primitives for computing the infinity norm of difference of pixels between two images.

### Basic NormDiff\_Inf

- **NppStatus nppiNormDiff\_Inf\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image NormDiff\_Inf.*
- **NppStatus nppiNormDiff\_Inf\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormDiff\_Inf ignoring alpha channel.*
- **NppStatus nppiNormDiff\_Inf\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormDiff\_Inf ignoring alpha channel.*
- **NppStatus nppiNormDiff\_Inf\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormDiff\_Inf ignoring alpha channel.*

- [NppStatus nppiNormDiff\\_Inf\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormDiff\_Inf ignoring alpha channel.*
- [NppStatus nppiNormDiff\\_Inf\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormDiff\_Inf.*
- [NppStatus nppiNormDiff\\_Inf\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormDiff\_Inf.*
- [NppStatus nppiNormDiff\\_Inf\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormDiff\_Inf.*
- [NppStatus nppiNormDiff\\_Inf\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormDiff\_Inf.*

## Masked NormDiff\_Inf

See [Masked Operation](#).

- [NppStatus nppiNormDiff\\_Inf\\_8u\\_C1MR](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 8-bit unsigned images NormDiff\_Inf.*
- [NppStatus nppiNormDiff\\_Inf\\_8s\\_C1MR](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 8-bit signed images NormDiff\_Inf.*
- [NppStatus nppiNormDiff\\_Inf\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 16-bit unsigned images NormDiff\_Inf.*
- [NppStatus nppiNormDiff\\_Inf\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 32-bit floating point images NormDiff\_Inf.*

## Masked Channel Mean

See [Masked Operation](#) and [Channel-of-Interest API](#).

- **NppStatus** **nppiNormDiff\_Inf\_8u\_C3CMR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image NormDiff\_Inf affecting only single channel.*

- **NppStatus** **nppiNormDiff\_Inf\_8s\_C3CMR** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 8-bit signed image NormDiff\_Inf affecting only single channel.*

- **NppStatus** **nppiNormDiff\_Inf\_16u\_C3CMR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image NormDiff\_Inf affecting only single channel.*

- **NppStatus** **nppiNormDiff\_Inf\_32f\_C3CMR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image NormDiff\_Inf affecting only single channel.*

## NormDiffInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff\_Inf primitives.

- **NppStatus** **nppiNormDiffInfGetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNormDiff\_Inf\_8u\_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNormDiff\_Inf\_16u\_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNormDiff\_Inf\_16s\_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNormDiff\_Inf\_32f\_C1R**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNormDiff\_Inf\_8u\_C1MR**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size for **nppiNormDiff\_Inf\_8s\_C1MR**.*
- **NppStatus** **nppiNormDiffInfGetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_16u\_C1MR*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_32f\\_C1MR](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_32f\_C1MR*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_8u\\_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_8u\_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_16u\\_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_16u\_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_16s\\_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_16s\_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_32f\_C3R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_8u\_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_16u\_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_16s\_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_32f\_C4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_8u\_AC4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_16u\_AC4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_16s\_AC4R*.

- **NppStatus** [nppiNormDiffInfGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) [oSizeROI](#), [int](#) \*hpBufferSize)

Buffer size for *nppiNormDiff\_Inf\_32f\_AC4R*.

- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiNormDiff\\_Inf\\_8u\\_C3CMR](#).*
- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiNormDiff\\_Inf\\_8s\\_C3CMR](#).*
- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiNormDiff\\_Inf\\_16u\\_C3CMR](#).*
- [NppStatus](#) [nppiNormDiffInfGetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Buffer size for [nppiNormDiff\\_Inf\\_32f\\_C3CMR](#).*

### 7.101.1 Detailed Description

Primitives for computing the infinity norm of difference of pixels between two images.

### 7.101.2 Function Documentation

**7.101.2.1** [NppStatus](#) [nppiNormDiff\\_Inf\\_16s\\_AC4R](#) ([const](#) [Npp16s](#) \* [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) \* [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) [aNormDiff](#)[3], [Npp8u](#) \* [pDeviceBuffer](#))

Four-channel 16-bit signed image NormDiff\_Inf ignoring alpha channel.

#### Parameters:

[pSrc1](#) [Source-Image Pointer](#).

[nSrc1Step](#) [Source-Image Line Step](#).

[pSrc2](#) [Source-Image Pointer](#).

[nSrc2Step](#) [Source-Image Line Step](#).

[oSizeROI](#) [Region-of-Interest \(ROI\)](#).

[aNormDiff](#) Array that contains computed Inf-norm of differences.

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.2** `NppStatus nppiNormDiff_Inf_16s_C1R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image NormDiff\_Inf.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.3** `NppStatus nppiNormDiff_Inf_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff\_Inf.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.4** `NppStatus nppiNormDiff_Inf_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff\_Inf.



**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormDiff* Array that contains computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffInfGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.5** `NppStatus nppiNormDiff_Inf_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff\_Inf ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormDiff* Array that contains computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffInfGetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.6** `NppStatus nppiNormDiff_Inf_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned images NormDiff\_Inf.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.7** `NppStatus nppiNormDiff_Inf_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff\_Inf.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.8** `NppStatus nppiNormDiff_Inf_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff\_Inf affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* [Channel\\_of\\_Interest](#) Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified.

**7.101.2.9 NppStatus nppiNormDiff\_Inf\_16u\_C3R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u \* pDeviceBuffer)**

Three-channel 16-bit unsigned image NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.10 NppStatus nppiNormDiff\_Inf\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u \* pDeviceBuffer)**

Four-channel 16-bit unsigned image NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.11** `NppStatus nppiNormDiff_Inf_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff\_Inf ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.101.2.12** `NppStatus nppiNormDiff_Inf_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point images NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.101.2.13** `NppStatus nppiNormDiff_Inf_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff\_Inf.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.101.2.14** `NppStatus nppiNormDiff_Inf_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff\_Inf affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.101.2.15** `NppStatus nppiNormDiff_Inf_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.101.2.16** `NppStatus nppiNormDiff_Inf_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.101.2.17** `NppStatus nppiNormDiff_Inf_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed images NormDiff\_Inf.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormDiff* Pointer to the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffInfGetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.18** `NppStatus nppiNormDiff_Inf_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff\_Inf affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pNormDiff* Pointer to the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffInfGetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.101.2.19** `NppStatus nppiNormDiff_Inf_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff\_Inf ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNormDiff* Array that contains computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffInfGetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.20** `NppStatus nppiNormDiff_Inf_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned images NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*pMask* [Mask-Image Pointer](#).  
*nMaskStep* [Mask-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pNormDiff* Pointer to the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffInfGetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.21** `NppStatus nppiNormDiff_Inf_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).



*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.22** `NppStatus nppiNormDiff_Inf_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff\_Inf affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.101.2.23** `NppStatus nppiNormDiff_Inf_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff\_Inf.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.24** `NppStatus nppiNormDiff_Inf_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffInfGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.101.2.25** `NppStatus nppiNormDiffInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Buffer size for [nppiNormDiff\\_Inf\\_16s\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

`NPP_NULL_POINTER_ERROR` if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.26 NppStatus nppiNormDiffInfGetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.27 NppStatus nppiNormDiffInfGetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.28 NppStatus nppiNormDiffInfGetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.29 NppStatus nppiNormDiffInfGetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_16u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.30 NppStatus nppiNormDiffInfGetBufferHostSize\_16u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNormDiff\\_Inf\\_16u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.31 NppStatus nppiNormDiffInfGetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNormDiff\\_Inf\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.32 NppStatus nppiNormDiffInfGetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNormDiff\\_Inf\\_16u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.33 NppStatus nppiNormDiffInfGetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.34 NppStatus nppiNormDiffInfGetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.35 NppStatus nppiNormDiffInfGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.36 NppStatus nppiNormDiffInfGetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_32f\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.37 NppStatus nppiNormDiffInfGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNormDiff\\_Inf\\_32f\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.38 NppStatus nppiNormDiffInfGetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNormDiff\\_Inf\\_32f\\_C3CMR](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.39 NppStatus nppiNormDiffInfGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size for [nppiNormDiff\\_Inf\\_32f\\_C3R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.40 NppStatus nppiNormDiffInfGetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.41 NppStatus nppiNormDiffInfGetBufferHostSize\_8s\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8s\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.42 NppStatus nppiNormDiffInfGetBufferHostSize\_8s\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8s\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.43 NppStatus nppiNormDiffInfGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.44 NppStatus nppiNormDiffInfGetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8u\\_C1MR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.45 NppStatus nppiNormDiffInfGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.46 NppStatus nppiNormDiffInfGetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8u\\_C3CMR](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.101.2.47 NppStatus nppiNormDiffInfGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.101.2.48 NppStatus nppiNormDiffInfGetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size for [nppiNormDiff\\_Inf\\_8u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.102 NormDiff\_L1

Primitives for computing the L1 norm of difference of pixels between two images.

### Basic NormDiff\_L1

- **NppStatus nppiNormDiff\_L1\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image NormDiff\_L1.*
- **NppStatus nppiNormDiff\_L1\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormDiff\_L1 ignoring alpha channel.*
- **NppStatus nppiNormDiff\_L1\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormDiff\_L1 ignoring alpha channel.*
- **NppStatus nppiNormDiff\_L1\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormDiff\_L1 ignoring alpha channel.*

- [NppStatus nppiNormDiff\\_L1\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) \*pDeviceBuffer)

*Four-channel 32-bit floating point image NormDiff\_L1 ignoring alpha channel.*

- [NppStatus nppiNormDiff\\_L1\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)

*Four-channel 8-bit unsigned image NormDiff\_L1.*

- [NppStatus nppiNormDiff\\_L1\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)

*Four-channel 16-bit unsigned image NormDiff\_L1.*

- [NppStatus nppiNormDiff\\_L1\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)

*Four-channel 16-bit signed image NormDiff\_L1.*

- [NppStatus nppiNormDiff\\_L1\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)

*Four-channel 32-bit floating point image NormDiff\_L1.*

## Masked NormDiff\_L1

See [Masked Operation](#).

- [NppStatus nppiNormDiff\\_L1\\_8u\\_C1MR](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked one-channel 8-bit unsigned image NormDiff\_L1.*

- [NppStatus nppiNormDiff\\_L1\\_8s\\_C1MR](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked one-channel 8-bit signed image NormDiff\_L1.*

- [NppStatus nppiNormDiff\\_L1\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked one-channel 16-bit unsigned image NormDiff\_L1.*

- [NppStatus nppiNormDiff\\_L1\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked one-channel 32-bit floating point image NormDiff\_L1.*

## Masked Channel NormDiff\_L1

See [Masked Operation](#) and [Channel-of-Interest API](#).

- [NppStatus nppiNormDiff\\_L1\\_8u\\_C3CMR](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image NormDiff\_L1 affecting only single channel.*

- [NppStatus nppiNormDiff\\_L1\\_8s\\_C3CMR](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked three-channel 8-bit signed image NormDiff\_L1 affecting only single channel.*

- [NppStatus nppiNormDiff\\_L1\\_16u\\_C3CMR](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image NormDiff\_L1 affecting only single channel.*

- [NppStatus nppiNormDiff\\_L1\\_32f\\_C3CMR](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, int nCOI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image NormDiff\_L1 affecting only single channel.*

## NormDiffL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff\_L1 primitives.

- [NppStatus nppiNormDiffL1GetBufferHostSize\\_8u\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_C1R.*
- [NppStatus nppiNormDiffL1GetBufferHostSize\\_16u\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_C1R.*
- [NppStatus nppiNormDiffL1GetBufferHostSize\\_16s\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_C1R.*
- [NppStatus nppiNormDiffL1GetBufferHostSize\\_32f\\_C1R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_C1R.*
- [NppStatus nppiNormDiffL1GetBufferHostSize\\_8u\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_C1MR.*
- [NppStatus nppiNormDiffL1GetBufferHostSize\\_8s\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8s\_C1MR.*
- [NppStatus nppiNormDiffL1GetBufferHostSize\\_16u\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_C1MR.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_32f\_C1MR** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_C1MR.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_8u\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_C3R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_16u\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_C3R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_16s\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_C3R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_32f\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_C3R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_8u\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_C4R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_16u\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_C4R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_16s\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_C4R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_32f\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_C4R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_8u\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_AC4R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_16u\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_AC4R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_16s\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_AC4R.*

- **NppStatus** **nppiNormDiffL1GetBufferHostSize\_32f\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_AC4R.*

- **NppStatus** [nppiNormDiffL1GetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L1\\_8u\\_C3CMR](#).*
- **NppStatus** [nppiNormDiffL1GetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L1\\_8s\\_C3CMR](#).*
- **NppStatus** [nppiNormDiffL1GetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L1\\_16u\\_C3CMR](#).*
- **NppStatus** [nppiNormDiffL1GetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L1\\_32f\\_C3CMR](#).*

### 7.102.1 Detailed Description

Primitives for computing the L1 norm of difference of pixels between two images.

### 7.102.2 Function Documentation

**7.102.2.1** **NppStatus** [nppiNormDiff\\_L1\\_16s\\_AC4R](#) ([const](#) [Npp16s](#) \* [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) \* [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) [aNormDiff](#)[3], [Npp8u](#) \* [pDeviceBuffer](#))

Four-channel 16-bit signed image NormDiff\_L1 ignoring alpha channel.

#### Parameters:

- [pSrc1](#) [Source-Image Pointer](#).
- [nSrc1Step](#) [Source-Image Line Step](#).
- [pSrc2](#) [Source-Image Pointer](#).
- [nSrc2Step](#) [Source-Image Line Step](#).
- [oSizeROI](#) [Region-of-Interest \(ROI\)](#).
- [aNormDiff](#) Array that contains computed Inf-norm of differences.
- [pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.2** **NppStatus** [nppiNormDiff\\_L1\\_16s\\_C1R](#) ([const](#) [Npp16s](#) \* [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) \* [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) \* [pNormDiff](#), [Npp8u](#) \* [pDeviceBuffer](#))

One-channel 16-bit signed image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormDiff* Pointer to the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL1GetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.3** `NppStatus nppiNormDiff_L1_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormDiff* Array that contains computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL1GetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.4** `NppStatus nppiNormDiff_L1_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.5** `NppStatus nppiNormDiff_L1_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff\_L1 ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.6** `NppStatus nppiNormDiff_L1_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormDiff\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.102.2.7** `NppStatus nppiNormDiff_L1_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.8** `NppStatus nppiNormDiff_L1_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff\_L1 affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.102.2.9** `NppStatus nppiNormDiff_L1_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.10** `NppStatus nppiNormDiff_L1_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.11** `NppStatus nppiNormDiff_L1_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff\_L1 ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormDiff* Array that contains computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL1GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.102.2.12** `NppStatus nppiNormDiff_L1_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormDiff* Pointer to the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL1GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.102.2.13** `NppStatus nppiNormDiff_L1_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.102.2.14** `NppStatus nppiNormDiff_L1_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff\_L1 affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.102.2.15** `NppStatus nppiNormDiff_L1_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff\_L1.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.102.2.16** `NppStatus nppiNormDiff_L1_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff\_L1.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.102.2.17** `NppStatus nppiNormDiff_L1_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormDiff\_L1.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.18** `NppStatus nppiNormDiff_L1_8s_C3CMR (const Npp8s *pSrc1, int nSrc1Step, const Npp8s *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f *pNormDiff, Npp8u *pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff\_L1 affecting only single channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.102.2.19** `NppStatus nppiNormDiff_L1_8u_AC4R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u *pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff\_L1 ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.20** `NppStatus nppiNormDiff_L1_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormDiff\_L1.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.21** `NppStatus nppiNormDiff_L1_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff\_L1.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.22** `NppStatus nppiNormDiff_L1_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff\_L1 affecting only single channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNormDiff* [Pointer to the computed Inf-norm of differences](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, \[Scratch Buffer and Host Pointer\]\(#\)](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.102.2.23** `NppStatus nppiNormDiff_L1_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* [Array that contains computed Inf-norm of differences](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, \[Scratch Buffer and Host Pointer\]\(#\)](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.102.2.24** `NppStatus nppiNormDiff_L1_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL1GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.102.2.25** `NppStatus nppiNormDiffL1GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.26** `NppStatus nppiNormDiffL1GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.27 NppStatus nppiNormDiffL1GetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.28 NppStatus nppiNormDiffL1GetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.29 NppStatus nppiNormDiffL1GetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.30 NppStatus nppiNormDiffL1GetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.31 NppStatus nppiNormDiffL1GetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.32 NppStatus nppiNormDiffL1GetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_16u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.33 NppStatus nppiNormDiffL1GetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.102.2.34 NppStatus nppiNormDiffL1GetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_16u\_C4R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.102.2.35 NppStatus nppiNormDiffL1GetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_AC4R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.102.2.36 NppStatus nppiNormDiffL1GetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_C1MR.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.102.2.37 NppStatus nppiNormDiffL1GetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_32f\_C1R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.38 NppStatus nppiNormDiffL1GetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_32f\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.39 NppStatus nppiNormDiffL1GetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_32f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.40 NppStatus nppiNormDiffL1GetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_32f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.41 NppStatus nppiNormDiffL1GetBufferHostSize\_8s\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.42 NppStatus nppiNormDiffL1GetBufferHostSize\_8s\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.43 NppStatus nppiNormDiffL1GetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.44 NppStatus nppiNormDiffL1GetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.45 NppStatus nppiNormDiffL1GetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.46 NppStatus nppiNormDiffL1GetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_8u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.47 NppStatus nppiNormDiffL1GetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L1\_8u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.102.2.48 NppStatus nppiNormDiffL1GetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L1\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



## 7.103 NormDiff\_L2

Primitives for computing the L2 norm of difference of pixels between two images.

### Basic NormDiff\_L2

- **NppStatus nppiNormDiff\_L2\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image NormDiff\_L2.*
- **NppStatus nppiNormDiff\_L2\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormDiff\_L2 ignoring alpha channel.*
- **NppStatus nppiNormDiff\_L2\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormDiff\_L2 ignoring alpha channel.*
- **NppStatus nppiNormDiff\_L2\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormDiff[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormDiff\_L2 ignoring alpha channel.*

- [NppStatus nppiNormDiff\\_L2\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[3], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormDiff\_L2 ignoring alpha channel.*
- [NppStatus nppiNormDiff\\_L2\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormDiff\_L2.*
- [NppStatus nppiNormDiff\\_L2\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormDiff\_L2.*
- [NppStatus nppiNormDiff\\_L2\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormDiff\_L2.*
- [NppStatus nppiNormDiff\\_L2\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormDiff[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormDiff\_L2.*

## Masked NormDiff\_L2

See [Masked Operation](#).

- [NppStatus nppiNormDiff\\_L2\\_8u\\_C1MR](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 8-bit unsigned image NormDiff\_L2.*
- [NppStatus nppiNormDiff\\_L2\\_8s\\_C1MR](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 8-bit signed image NormDiff\_L2.*
- [NppStatus nppiNormDiff\\_L2\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 16-bit unsigned image NormDiff\_L2.*
- [NppStatus nppiNormDiff\\_L2\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormDiff, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 32-bit floating point image NormDiff\_L2.*

## Masked Channel NormDiff\_L2

See [Masked Operation](#) and [Channel-of-Interest API](#).

- **NppStatus** **nppiNormDiff\_L2\_8u\_C3CMR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image NormDiff\_L2 affecting only single channel.*

- **NppStatus** **nppiNormDiff\_L2\_8s\_C3CMR** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 8-bit signed image NormDiff\_L2 affecting only single channel.*

- **NppStatus** **nppiNormDiff\_L2\_16u\_C3CMR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image NormDiff\_L2 affecting only single channel.*

- **NppStatus** **nppiNormDiff\_L2\_32f\_C3CMR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormDiff, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image NormDiff\_L2 affecting only single channel.*

## NormDiffL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormDiff\_L2 primitives.

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_C1R.*
- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_C1R.*
- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_C1R.*
- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_C1R.*
- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_C1MR.*
- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8s\_C1MR.*
- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_C1MR.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_32f\_C1MR** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_C1MR.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_8u\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_C3R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16u\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_C3R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16s\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_C3R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_32f\_C3R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_C3R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_8u\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_C4R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16u\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_C4R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16s\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_C4R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_32f\_C4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_C4R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_8u\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_AC4R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16u\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_AC4R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_16s\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_AC4R.*

- **NppStatus** **nppiNormDiffL2GetBufferHostSize\_32f\_AC4R** (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_AC4R.*

- **NppStatus** [nppiNormDiffL2GetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L2\\_8u\\_C3CMR](#).*
- **NppStatus** [nppiNormDiffL2GetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L2\\_8s\\_C3CMR](#).*
- **NppStatus** [nppiNormDiffL2GetBufferHostSize\\_16u\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L2\\_16u\\_C3CMR](#).*
- **NppStatus** [nppiNormDiffL2GetBufferHostSize\\_32f\\_C3CMR](#) ([NppiSize](#) [oSizeROI](#), [int](#) [\\*hpBufferSize](#))  
*Computes the device scratch buffer size (in bytes) for [nppiNormDiff\\_L2\\_32f\\_C3CMR](#).*

### 7.103.1 Detailed Description

Primitives for computing the L2 norm of difference of pixels between two images.

### 7.103.2 Function Documentation

**7.103.2.1** **NppStatus** [nppiNormDiff\\_L2\\_16s\\_AC4R](#) ([const](#) [Npp16s](#) \* [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) \* [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) [aNormDiff](#)[3], [Npp8u](#) \* [pDeviceBuffer](#))

Four-channel 16-bit signed image NormDiff\_L2 ignoring alpha channel.

#### Parameters:

- [pSrc1](#) [Source-Image Pointer](#).
- [nSrc1Step](#) [Source-Image Line Step](#).
- [pSrc2](#) [Source-Image Pointer](#).
- [nSrc2Step](#) [Source-Image Line Step](#).
- [oSizeROI](#) [Region-of-Interest \(ROI\)](#).
- [aNormDiff](#) Array that contains computed Inf-norm of differences.
- [pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.2** **NppStatus** [nppiNormDiff\\_L2\\_16s\\_C1R](#) ([const](#) [Npp16s](#) \* [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) \* [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) \* [pNormDiff](#), [Npp8u](#) \* [pDeviceBuffer](#))

One-channel 16-bit signed image NormDiff\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pNormDiff* Pointer to the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL2GetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.3** `NppStatus nppiNormDiff_L2_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormDiff\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNormDiff* Array that contains computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL2GetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.4** `NppStatus nppiNormDiff_L2_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormDiff\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.5** `NppStatus nppiNormDiff_L2_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff\_L2 ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.6** `NppStatus nppiNormDiff_L2_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.7** `NppStatus nppiNormDiff_L2_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.8** `NppStatus nppiNormDiff_L2_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormDiff\_L2 affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.



**7.103.2.9** `NppStatus nppiNormDiff_L2_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.10** `NppStatus nppiNormDiff_L2_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.11** `NppStatus nppiNormDiff_L2_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff\_L2 ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormDiff* Array that contains computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL2GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.103.2.12** `NppStatus nppiNormDiff_L2_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormDiff\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormDiff* Pointer to the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormDiffL2GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.103.2.13** `NppStatus nppiNormDiff_L2_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormDiff\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.103.2.14** `NppStatus nppiNormDiff_L2_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormDiff\_L2 affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.103.2.15** `NppStatus nppiNormDiff_L2_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormDiff\_L2.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.103.2.16** `NppStatus nppiNormDiff_L2_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormDiff\_L2.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.103.2.17** `NppStatus nppiNormDiff_L2_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormDiff\_L2.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.18** `NppStatus nppiNormDiff_L2_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormDiff\_L2 affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.103.2.19** `NppStatus nppiNormDiff_L2_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff\_L2 ignoring alpha channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.20** `NppStatus nppiNormDiff_L2_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.21** `NppStatus nppiNormDiff_L2_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.22** `NppStatus nppiNormDiff_L2_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormDiff, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormDiff\_L2 affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormDiff* Pointer to the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified.

**7.103.2.23** `NppStatus nppiNormDiff_L2_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.24** `NppStatus nppiNormDiff_L2_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormDiff[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormDiff\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormDiff* Array that contains computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormDiffL2GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.103.2.25** `NppStatus nppiNormDiffL2GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.26** `NppStatus nppiNormDiffL2GetBufferHostSize_16s_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.103.2.27 NppStatus nppiNormDiffL2GetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.28 NppStatus nppiNormDiffL2GetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.29 NppStatus nppiNormDiffL2GetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.30 NppStatus nppiNormDiffL2GetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.31 NppStatus nppiNormDiffL2GetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.32 NppStatus nppiNormDiffL2GetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_16u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.33 NppStatus nppiNormDiffL2GetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.34 NppStatus nppiNormDiffL2GetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.35 NppStatus nppiNormDiffL2GetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.36 NppStatus nppiNormDiffL2GetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.37 NppStatus nppiNormDiffL2GetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.38 NppStatus nppiNormDiffL2GetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_32f\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.39 NppStatus nppiNormDiffL2GetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_32f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.40 NppStatus nppiNormDiffL2GetBufferHostSize\_32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_32f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.41 NppStatus nppiNormDiffL2GetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.42 NppStatus nppiNormDiffL2GetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.43 NppStatus nppiNormDiffL2GetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.44 NppStatus nppiNormDiffL2GetBufferHostSize\_8u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.45 NppStatus nppiNormDiffL2GetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.46 NppStatus nppiNormDiffL2GetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_8u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.47 NppStatus nppiNormDiffL2GetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormDiff\_L2\_8u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.103.2.48 NppStatus nppiNormDiffL2GetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormDiff\_L2\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.104 NormRel\_Inf

Primitives for computing the relative error of infinity norm between two images.

### Basic NormRel\_Inf

- **NppStatus nppiNormRel\_Inf\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image NormRel\_Inf.*
- **NppStatus nppiNormRel\_Inf\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormRel\_Inf ignoring alpha channel.*
- **NppStatus nppiNormRel\_Inf\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormRel\_Inf ignoring alpha channel.*
- **NppStatus nppiNormRel\_Inf\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormRel\_Inf ignoring alpha channel.*



- **NppStatus** **nppiNormRel\_Inf\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit floating point image NormRel\_Inf ignoring alpha channel.*

- **NppStatus** **nppiNormRel\_Inf\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[4], **Npp8u** \*pDeviceBuffer)

*Four-channel 8-bit unsigned image NormRel\_Inf.*

- **NppStatus** **nppiNormRel\_Inf\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[4], **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit unsigned image NormRel\_Inf.*

- **NppStatus** **nppiNormRel\_Inf\_16s\_C4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[4], **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit signed image NormRel\_Inf.*

- **NppStatus** **nppiNormRel\_Inf\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[4], **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit floating point image NormRel\_Inf.*

## Masked NormRel\_Inf

See [Masked Operation](#).

- **NppStatus** **nppiNormRel\_Inf\_8u\_C1MR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked one-channel 8-bit unsigned image NormRel\_Inf.*

- **NppStatus** **nppiNormRel\_Inf\_8s\_C1MR** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked one-channel 8-bit signed image NormRel\_Inf.*

- **NppStatus** **nppiNormRel\_Inf\_16u\_C1MR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked one-channel 16-bit unsigned image NormRel\_Inf.*

- **NppStatus** **nppiNormRel\_Inf\_32f\_C1MR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked one-channel 32-bit floating point image NormRel\_Inf.*

## Masked Channel NormRel\_Inf

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormRel_Inf_8u_C3CMR` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image NormRel\_Inf affecting only single channel.*

- `NppStatus nppiNormRel_Inf_8s_C3CMR` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 8-bit signed image NormRel\_Inf affecting only single channel.*

- `NppStatus nppiNormRel_Inf_16u_C3CMR` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image NormRel\_Inf affecting only single channel.*

- `NppStatus nppiNormRel_Inf_32f_C3CMR` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image NormRel\_Inf affecting only single channel.*

## NormRelInfGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel\_Inf primitives.

- `NppStatus nppiNormRelInfGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_C1R.*
- `NppStatus nppiNormRelInfGetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_C1R.*
- `NppStatus nppiNormRelInfGetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16s\_C1R.*
- `NppStatus nppiNormRelInfGetBufferHostSize_32s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32s\_C1R.*
- `NppStatus nppiNormRelInfGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C1R.*
- `NppStatus nppiNormRelInfGetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_C1MR.*
- `NppStatus nppiNormRelInfGetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8s\_C1MR.*
- `NppStatus nppiNormRelInfGetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_C1MR.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_32f\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C1MR.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_8u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_C3R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_16u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_C3R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_16s\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16s\_C3R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C3R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_8u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_C4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_16u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_C4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16s\_C4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_8u\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_AC4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_16u\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_AC4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16s\_AC4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_AC4R.*

- **NppStatus** **nppiNormRelInfGetBufferHostSize\_8u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_C3CMR.*

- **NppStatus** [nppiNormRelInfGetBufferHostSize\\_8s\\_C3CMR](#) (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for [nppiNormRel\\_Inf\\_8s\\_C3CMR](#).*
- **NppStatus** [nppiNormRelInfGetBufferHostSize\\_16u\\_C3CMR](#) (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for [nppiNormRel\\_Inf\\_16u\\_C3CMR](#).*
- **NppStatus** [nppiNormRelInfGetBufferHostSize\\_32f\\_C3CMR](#) (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for [nppiNormRel\\_Inf\\_32f\\_C3CMR](#).*

### 7.104.1 Detailed Description

Primitives for computing the relative error of infinity norm between two images.

### 7.104.2 Function Documentation

**7.104.2.1** **NppStatus** [nppiNormRel\\_Inf\\_16s\\_AC4R](#) (const **Npp16s** \* *pSrc1*, int *nSrc1Step*, const **Npp16s** \* *pSrc2*, int *nSrc2Step*, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \* *pDeviceBuffer*)

Four-channel 16-bit signed image NormRel\_Inf ignoring alpha channel.

#### Parameters:

- pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNormRel* Array that contains the computed relative error for the infinity norm of two images.  
*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelInfGetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.2** **NppStatus** [nppiNormRel\\_Inf\\_16s\\_C1R](#) (const **Npp16s** \* *pSrc1*, int *nSrc1Step*, const **Npp16s** \* *pSrc2*, int *nSrc2Step*, **NppiSize** oSizeROI, **Npp64f** \* *pNormRel*, **Npp8u** \* *pDeviceBuffer*)

One-channel 16-bit signed image NormRel\_Inf.

#### Parameters:

- pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.3** `NppStatus nppiNormRel_Inf_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel\_Inf.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.4** `NppStatus nppiNormRel_Inf_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel\_Inf.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.5** `NppStatus nppiNormRel_Inf_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel\_Inf ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.6** `NppStatus nppiNormRel_Inf_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.7** `NppStatus nppiNormRel_Inf_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.8** `NppStatus nppiNormRel_Inf_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel\_Inf affecting only single channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.9** `NppStatus nppiNormRel_Inf_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNormRel* Array that contains the computed relative error for the infinity norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.10** `NppStatus nppiNormRel_Inf_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNormRel* Array that contains the computed relative error for the infinity norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.



**7.104.2.11 NppStatus nppiNormRel\_Inf\_32f\_AC4R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[3], Npp8u \* *pDeviceBuffer*)

Four-channel 32-bit floating point image NormRel\_Inf ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.12 NppStatus nppiNormRel\_Inf\_32f\_C1MR** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, const Npp8u \* *pMask*, int *nMaskStep*, NppiSize *oSizeROI*, Npp64f \* *pNormRel*, Npp8u \* *pDeviceBuffer*)

Masked one-channel 32-bit floating point image NormRel\_Inf.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.13** `NppStatus nppiNormRel_Inf_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.14** `NppStatus nppiNormRel_Inf_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel\_Inf affecting only single channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the infinity norm of the second image is zero.

**7.104.2.15** `NppStatus nppiNormRel_Inf_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.16** `NppStatus nppiNormRel_Inf_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.17** `NppStatus nppiNormRel_Inf_8s_C1MR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit signed image NormRel\_Inf.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- pNormRel* Pointer to the computed relative error for the infinity norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.18** `NppStatus nppiNormRel_Inf_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel\_Inf affecting only single channel.

**Parameters:**

- pSrc1* Source-Image Pointer.
- nSrc1Step* Source-Image Line Step.
- pSrc2* Source-Image Pointer.
- nSrc2Step* Source-Image Line Step.
- pMask* Mask-Image Pointer.
- nMaskStep* Mask-Image Line Step.
- oSizeROI* Region-of-Interest (ROI).
- nCOI* Channel\_of\_Interest Number.
- pNormRel* Pointer to the computed relative error for the infinity norm of two images.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.19** `NppStatus nppiNormRel_Inf_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel\_Inf ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.20** `NppStatus nppiNormRel_Inf_8u_C1MR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.21** `NppStatus nppiNormRel_Inf_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.22** `NppStatus nppiNormRel_Inf_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel\_Inf affecting only single channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNormRel* Pointer to the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.23** `NppStatus nppiNormRel_Inf_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.24** `NppStatus nppiNormRel_Inf_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel\_Inf.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the infinity norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelInfGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the infinity norm of the second image is zero.

**7.104.2.25** `NppStatus nppiNormRelInfGetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for `nppiNormRel_Inf_16s_AC4R`.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.26 NppStatus nppiNormRelInfGetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_16s\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.27 NppStatus nppiNormRelInfGetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_16s\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.28 NppStatus nppiNormRelInfGetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_16s\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.104.2.29 NppStatus nppiNormRelInfGetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.30 NppStatus nppiNormRelInfGetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.31 NppStatus nppiNormRelInfGetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.32 NppStatus nppiNormRelInfGetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_16u\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.33 NppStatus nppiNormRelInfGetBufferHostSize\_16u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_16u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.34 NppStatus nppiNormRelInfGetBufferHostSize\_16u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_16u\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.35 NppStatus nppiNormRelInfGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_32f\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.36 NppStatus nppiNormRelInfGetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.37 NppStatus nppiNormRelInfGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.38 NppStatus nppiNormRelInfGetBufferHostSize\_32f\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.39 NppStatus nppiNormRelInfGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.40 NppStatus nppiNormRelInfGetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_32f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.41 NppStatus nppiNormRelInfGetBufferHostSize\_32s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_32s\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.42 NppStatus nppiNormRelInfGetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_8s\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.43 NppStatus nppiNormRelInfGetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.44 NppStatus nppiNormRelInfGetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.45 NppStatus nppiNormRelInfGetBufferHostSize\_8u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.46 NppStatus nppiNormRelInfGetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_Inf\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.47 NppStatus nppiNormRelInfGetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_8u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.48 NppStatus nppiNormRelInfGetBufferHostSize\_8u\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_8u\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.104.2.49 NppStatus nppiNormRelInfGetBufferHostSize\_8u\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_Inf\_8u\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.105 NormRel\_L1

Primitives for computing the relative error of L1 norm between two images.

### Basic NormRel\_L1

- **NppStatus nppiNormRel\_L1\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image NormRel\_L1.*
- **NppStatus nppiNormRel\_L1\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit signed image NormRel\_L1 ignoring alpha channel.*
- **NppStatus nppiNormRel\_L1\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormRel\_L1 ignoring alpha channel.*
- **NppStatus nppiNormRel\_L1\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormRel\_L1 ignoring alpha channel.*

- [NppStatus nppiNormRel\\_L1\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[3], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormRel\_L1 ignoring alpha channel.*
- [NppStatus nppiNormRel\\_L1\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormRel\_L1.*
- [NppStatus nppiNormRel\\_L1\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormRel\_L1.*
- [NppStatus nppiNormRel\\_L1\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormRel\_L1.*
- [NppStatus nppiNormRel\\_L1\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormRel\_L1.*

## Masked NormRel\_L1

See [Masked Operation](#).

- [NppStatus nppiNormRel\\_L1\\_8u\\_C1MR](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*One-channel 8-bit unsigned image NormRel\_L1.*
- [NppStatus nppiNormRel\\_L1\\_8s\\_C1MR](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*One-channel 8-bit signed image NormRel\_L1.*
- [NppStatus nppiNormRel\\_L1\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*One-channel 16-bit unsigned image NormRel\_L1.*
- [NppStatus nppiNormRel\\_L1\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*One-channel 32-bit floating point image NormRel\_L1.*

## Masked Channel NormRel\_L1

See [Masked Operation](#) and [Channel-of-Interest API](#).



- **NppStatus** **nppiNormRel\_L1\_8u\_C3CMR** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image NormRel\_L1 affecting only single channel.*

- **NppStatus** **nppiNormRel\_L1\_8s\_C3CMR** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 8-bit signed image NormRel\_L1 affecting only single channel.*

- **NppStatus** **nppiNormRel\_L1\_16u\_C3CMR** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image NormRel\_L1 affecting only single channel.*

- **NppStatus** **nppiNormRel\_L1\_32f\_C3CMR** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, const **Npp8u** \*pMask, int nMaskStep, **NppiSize** oSizeROI, int nCOI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image NormRel\_L1 affecting only single channel.*

## NormRelL1GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel\_L1 primitives.

- **NppStatus** **nppiNormRelL1GetBufferHostSize\_8u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_C1R.*

- **NppStatus** **nppiNormRelL1GetBufferHostSize\_16u\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_C1R.*

- **NppStatus** **nppiNormRelL1GetBufferHostSize\_16s\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_C1R.*

- **NppStatus** **nppiNormRelL1GetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C1R.*

- **NppStatus** **nppiNormRelL1GetBufferHostSize\_8u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_C1MR.*

- **NppStatus** **nppiNormRelL1GetBufferHostSize\_8s\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8s\_C1MR.*

- **NppStatus** **nppiNormRelL1GetBufferHostSize\_16u\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_C1MR.*

- [NppStatus nppiNormRelL1GetBufferHostSize\\_32f\\_C1MR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C1MR.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_8u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_C3R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_16u\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_C3R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_16s\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_C3R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C3R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_8u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_C4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_16u\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_C4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_16s\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_C4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_32f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_AC4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_16u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_AC4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_16s\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_AC4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_AC4R.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_8u\\_C3CMR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_C3CMR.*
- [NppStatus nppiNormRelL1GetBufferHostSize\\_8s\\_C3CMR](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8s\_C3CMR.*

- **NppStatus** [nppiNormRelL1GetBufferHostSize\\_16u\\_C3CMR](#) (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_C3CMR.*

- **NppStatus** [nppiNormRelL1GetBufferHostSize\\_32f\\_C3CMR](#) (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C3CMR.*

## 7.105.1 Detailed Description

Primitives for computing the relative error of L1 norm between two images.

## 7.105.2 Function Documentation

**7.105.2.1** **NppStatus** [nppiNormRel\\_L1\\_16s\\_AC4R](#) (**const** **Npp16s** \* *pSrc1*, **int** *nSrc1Step*, **const** **Npp16s** \* *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oSizeROI*, **Npp64f** *aNormRel*[3], **Npp8u** \* *pDeviceBuffer*)

Four-channel 16-bit signed image NormRel\_L1 ignoring alpha channel.

### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or **NPP\_DIVISOR\_ERROR** if the L1 norm of the second image is zero.

**7.105.2.2** **NppStatus** [nppiNormRel\\_L1\\_16s\\_C1R](#) (**const** **Npp16s** \* *pSrc1*, **int** *nSrc1Step*, **const** **Npp16s** \* *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oSizeROI*, **Npp64f** \* *pNormRel*, **Npp8u** \* *pDeviceBuffer*)

One-channel 16-bit signed image NormRel\_L1.

### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppiNormRelL1GetBufferHostSize_16s_C1R` to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

**7.105.2.3** `NppStatus nppiNormRel_L1_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel\_L1.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `nppiNormRelL1GetBufferHostSize_16s_C3R` to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

**7.105.2.4** `NppStatus nppiNormRel_L1_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel\_L1.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.5** `NppStatus nppiNormRel_L1_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel\_L1 ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.6** `NppStatus nppiNormRel_L1_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

**7.105.2.7** `NppStatus nppiNormRel_L1_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

**7.105.2.8** `NppStatus nppiNormRel_L1_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel\_L1 affecting only single channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNormRel* Pointer to the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

**7.105.2.9** `NppStatus nppiNormRel_L1_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image NormRel\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

**7.105.2.10** `NppStatus nppiNormRel_L1_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L1 norm of the second image is zero.

**7.105.2.11** `NppStatus nppiNormRel_L1_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel\_L1 ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormRel* Array that contains the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.12** `NppStatus nppiNormRel_L1_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormRel* Pointer to the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.13** `NppStatus nppiNormRel_L1_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel\_L1.



**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormRel* Pointer to the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.14** `NppStatus nppiNormRel_L1_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel\_L1 affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pNormRel* Pointer to the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified, NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.15** `NppStatus nppiNormRel_L1_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNormRel* Array that contains the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.16** `NppStatus nppiNormRel_L1_32f_C4R (const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u *pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aNormRel* Array that contains the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.17** `NppStatus nppiNormRel_L1_8s_C1MR (const Npp8s *pSrc1, int nSrc1Step, const Npp8s *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

One-channel 8-bit signed image NormRel\_L1.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.18** `NppStatus nppiNormRel_L1_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel\_L1 affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormRel* Pointer to the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL1GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.19** `NppStatus nppiNormRel_L1_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image NormRel\_L1 ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormRel* Array that contains the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.20** `NppStatus nppiNormRel_L1_8u_C1MR (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormRel* Pointer to the computed relative error for the L1 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL1GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.21** `NppStatus nppiNormRel_L1_8u_C1R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel\_L1.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.22** `NppStatus nppiNormRel_L1_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel\_L1 affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormRel* Pointer to the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.23** `NppStatus nppiNormRel_L1_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel\_L1.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.24** `NppStatus nppiNormRel_L1_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel\_L1.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L1 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL1GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L1 norm of the second image is zero.

**7.105.2.25** `NppStatus nppiNormRelL1GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_AC4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.26 NppStatus nppiNormRelL1GetBufferHostSize\_16s\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.27 NppStatus nppiNormRelL1GetBufferHostSize\_16s\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.28 NppStatus nppiNormRelL1GetBufferHostSize\_16s\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.29 NppStatus nppiNormRelL1GetBufferHostSize\_16u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.30 NppStatus nppiNormRelL1GetBufferHostSize\_16u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L1\_16u\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.31 NppStatus nppiNormRelL1GetBufferHostSize\_16u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L1\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.32 NppStatus nppiNormRelL1GetBufferHostSize\_16u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L1\_16u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.105.2.33 NppStatus nppiNormRelL1GetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.34 NppStatus nppiNormRelL1GetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.35 NppStatus nppiNormRelL1GetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.36 NppStatus nppiNormRelL1GetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.37 NppStatus nppiNormRelL1GetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C1R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.38 NppStatus nppiNormRelL1GetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C3CMR.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.39 NppStatus nppiNormRelL1GetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C3R.

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.40 NppStatus nppiNormRelL1GetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.41 NppStatus nppiNormRelL1GetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.42 NppStatus nppiNormRelL1GetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.43 NppStatus nppiNormRelL1GetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.44 NppStatus nppiNormRelL1GetBufferHostSize\_8u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L1\_8u\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.45 NppStatus nppiNormRelL1GetBufferHostSize\_8u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L1\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.46 NppStatus nppiNormRelL1GetBufferHostSize\_8u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L1\_8u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.47 NppStatus nppiNormRelL1GetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.105.2.48 NppStatus nppiNormRelL1GetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L1\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.106 NormRel\_L2

Primitives for computing the relative error of L2 norm between two images.

### Basic NormRel\_L2

- **NppStatus nppiNormRel\_L2\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pNormRel, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image NormRel\_L2.*
- **NppStatus nppiNormRel\_L2\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormRel\_L2 ignoring alpha channel.*
- **NppStatus nppiNormRel\_L2\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormRel\_L2 ignoring alpha channel.*
- **NppStatus nppiNormRel\_L2\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aNormRel[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormRel\_L2 ignoring alpha channel.*

- [NppStatus nppiNormRel\\_L2\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[3], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormRel\_L2 ignoring alpha channel.*
- [NppStatus nppiNormRel\\_L2\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image NormRel\_L2.*
- [NppStatus nppiNormRel\\_L2\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image NormRel\_L2.*
- [NppStatus nppiNormRel\\_L2\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit signed image NormRel\_L2.*
- [NppStatus nppiNormRel\\_L2\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) aNormRel[4], [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 32-bit floating point image NormRel\_L2.*

## Masked NormRel\_L2

See [Masked Operation](#).

- [NppStatus nppiNormRel\\_L2\\_8u\\_C1MR](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 8-bit unsigned image NormRel\_L2.*
- [NppStatus nppiNormRel\\_L2\\_8s\\_C1MR](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 8-bit signed image NormRel\_L2.*
- [NppStatus nppiNormRel\\_L2\\_16u\\_C1MR](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 16-bit unsigned image NormRel\_L2.*
- [NppStatus nppiNormRel\\_L2\\_32f\\_C1MR](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, const [Npp8u](#) \*pMask, int nMaskStep, [NppiSize](#) oSizeROI, [Npp64f](#) \*pNormRel, [Npp8u](#) \*pDeviceBuffer)  
*Masked one-channel 32-bit floating point image NormRel\_L2.*

## Masked Channel NormRel\_L2

See [Masked Operation](#) and [Channel-of-Interest API](#).

- `NppStatus nppiNormRel_L2_8u_C3CMR` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 8-bit unsigned image NormRel\_L2 affecting only single channel.*

- `NppStatus nppiNormRel_L2_8s_C3CMR` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 8-bit signed image NormRel\_L2 affecting only single channel.*

- `NppStatus nppiNormRel_L2_16u_C3CMR` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 16-bit unsigned image NormRel\_L2 affecting only single channel.*

- `NppStatus nppiNormRel_L2_32f_C3CMR` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, const `Npp8u` \*pMask, int nMaskStep, `NppiSize` oSizeROI, int nCOI, `Npp64f` \*pNormRel, `Npp8u` \*pDeviceBuffer)

*Masked three-channel 32-bit floating point image NormRel\_L2 affecting only single channel.*

## NormRelL2GetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the NormRel\_L2 primitives.

- `NppStatus nppiNormRelL2GetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_C1R.*

- `NppStatus nppiNormRelL2GetBufferHostSize_16u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_C1R.*

- `NppStatus nppiNormRelL2GetBufferHostSize_16s_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_C1R.*

- `NppStatus nppiNormRelL2GetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_C1R.*

- `NppStatus nppiNormRelL2GetBufferHostSize_8u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_C1MR.*

- `NppStatus nppiNormRelL2GetBufferHostSize_8s_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8s\_C1MR.*

- `NppStatus nppiNormRelL2GetBufferHostSize_16u_C1MR` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_C1MR.*



- **NppStatus** **nppiNormRelL2GetBufferHostSize\_32f\_C1MR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_C1MR.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_8u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_C3R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_16u\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_C3R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_16s\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_C3R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_C3R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_8u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_C4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_16u\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_C4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_16s\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_C4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_C4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_8u\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_AC4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_16u\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_AC4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_16s\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_AC4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_AC4R.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_8u\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_C3CMR.*
- **NppStatus** **nppiNormRelL2GetBufferHostSize\_8s\_C3CMR** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8s\_C3CMR.*

- **NppStatus** [nppiNormRelL2GetBufferHostSize\\_16u\\_C3CMR](#) (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_C3CMR.*

- **NppStatus** [nppiNormRelL2GetBufferHostSize\\_32f\\_C3CMR](#) (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_C3CMR.*

### 7.106.1 Detailed Description

Primitives for computing the relative error of L2 norm between two images.

### 7.106.2 Function Documentation

**7.106.2.1** **NppStatus** [nppiNormRel\\_L2\\_16s\\_AC4R](#) (**const** **Npp16s** \* *pSrc1*, **int** *nSrc1Step*, **const** **Npp16s** \* *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oSizeROI*, **Npp64f** *aNormRel*[3], **Npp8u** \* *pDeviceBuffer*)

Four-channel 16-bit signed image NormRel\_L2 ignoring alpha channel.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16s\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

**7.106.2.2** **NppStatus** [nppiNormRel\\_L2\\_16s\\_C1R](#) (**const** **Npp16s** \* *pSrc1*, **int** *nSrc1Step*, **const** **Npp16s** \* *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oSizeROI*, **Npp64f** \* *pNormRel*, **Npp8u** \* *pDeviceBuffer*)

One-channel 16-bit signed image NormRel\_L2.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.3** `NppStatus nppiNormRel_L2_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image NormRel\_L2.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.4** `NppStatus nppiNormRel_L2_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image NormRel\_L2.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.5** `NppStatus nppiNormRel_L2_16u_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image NormRel\_L2 ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.6** `NppStatus nppiNormRel_L2_16u_C1MR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 16-bit unsigned image NormRel\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.7** `NppStatus nppiNormRel_L2_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image NormRel\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.8** `NppStatus nppiNormRel_L2_16u_C3CMR (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 16-bit unsigned image NormRel\_L2 affecting only single channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pMask* [Mask-Image Pointer](#).

*nMaskStep* [Mask-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nCOI* [Channel\\_of\\_Interest Number](#).

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16u\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.9 NppStatus nppiNormRel\_L2\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[3], Npp8u \* *pDeviceBuffer*)**

Three-channel 16-bit unsigned image NormRel\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.10 NppStatus nppiNormRel\_L2\_16u\_C4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[4], Npp8u \* *pDeviceBuffer*)**

Four-channel 16-bit unsigned image NormRel\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.11 NppStatus nppiNormRel\_L2\_32f\_AC4R (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f *aNormRel*[3], Npp8u \* *pDeviceBuffer*)**

Four-channel 32-bit floating point image NormRel\_L2 ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormRel* Array that contains the computed relative error for the L2 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL2GetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.12** `NppStatus nppiNormRel_L2_32f_C1MR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked one-channel 32-bit floating point image NormRel\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormRel* Pointer to the computed relative error for the L2 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL2GetBufferHostSize\\_32f\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.13** `NppStatus nppiNormRel_L2_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image NormRel\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormRel* Pointer to the computed relative error for the L2 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL2GetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.14** `NppStatus nppiNormRel_L2_32f_C3CMR (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 32-bit floating point image NormRel\_L2 affecting only single channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nCOI* Channel\_of\_Interest Number.  
*pNormRel* Pointer to the computed relative error for the L2 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL2GetBufferHostSize\\_32f\\_C3CMR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.15** `NppStatus nppiNormRel_L2_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image NormRel\_L2.



**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.16** `NppStatus nppiNormRel_L2_32f_C4R (const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u *pDeviceBuffer)`

Four-channel 32-bit floating point image NormRel\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.17** `NppStatus nppiNormRel_L2_8s_C1MR (const Npp8s *pSrc1, int nSrc1Step, const Npp8s *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

Masked one-channel 8-bit signed image NormRel\_L2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize\\_8s\\_C1MR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.18** `NppStatus nppiNormRel_L2_8s_C3CMR (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit signed image NormRel\_L2 affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiNormRelL2GetBufferHostSize\\_8s\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_COI\_ERROR if an invalid channel of interest is specified, or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.19** `NppStatus nppiNormRel_L2_8u_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel\_L2 ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*aNormRel* Array that contains the computed relative error for the L2 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL2GetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.20** `NppStatus nppiNormRel_L2_8u_C1MR (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, const Npp8u *pMask, int nMaskStep, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

Masked one-channel 8-bit unsigned image NormRel\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pMask* Mask-Image Pointer.  
*nMaskStep* Mask-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pNormRel* Pointer to the computed relative error for the L2 norm of two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiNormRelL2GetBufferHostSize\\_8u\\_C1MR](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.21** `NppStatus nppiNormRel_L2_8u_C1R (const Npp8u *pSrc1, int nSrc1Step, const Npp8u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pNormRel, Npp8u *pDeviceBuffer)`

One-channel 8-bit unsigned image NormRel\_L2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

**7.106.2.22** `NppStatus nppiNormRel_L2_8u_C3CMR (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, const Npp8u * pMask, int nMaskStep, NppiSize oSizeROI, int nCOI, Npp64f * pNormRel, Npp8u * pDeviceBuffer)`

Masked three-channel 8-bit unsigned image NormRel\_L2 affecting only single channel.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pMask* Mask-Image Pointer.

*nMaskStep* Mask-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nCOI* Channel\_of\_Interest Number.

*pNormRel* Pointer to the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_8u\\_C3CMR](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_COI_ERROR` if an invalid channel of interest is specified, or `NPP_DIVISOR_ERROR` if the L2 norm of the second image is zero.

**7.106.2.23** `NppStatus nppiNormRel_L2_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image NormRel\_L2.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.24** `NppStatus nppiNormRel_L2_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aNormRel[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image NormRel\_L2.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aNormRel* Array that contains the computed relative error for the L2 norm of two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiNormRelL2GetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_DIVISOR\_ERROR if the L2 norm of the second image is zero.

**7.106.2.25** `NppStatus nppiNormRelL2GetBufferHostSize_16s_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_AC4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.106.2.26 NppStatus nppiNormRelL2GetBufferHostSize\_16s\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_C1R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.106.2.27 NppStatus nppiNormRelL2GetBufferHostSize\_16s\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_C3R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.106.2.28 NppStatus nppiNormRelL2GetBufferHostSize\_16s\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16s\_C4R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

#### 7.106.2.29 NppStatus nppiNormRelL2GetBufferHostSize\_16u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_AC4R.

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.30 NppStatus nppiNormRelL2GetBufferHostSize\_16u\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_16u\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.31 NppStatus nppiNormRelL2GetBufferHostSize\_16u\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_16u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.32 NppStatus nppiNormRelL2GetBufferHostSize\_16u\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_16u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.106.2.33 NppStatus nppiNormRelL2GetBufferHostSize\_16u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_C3R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.106.2.34 NppStatus nppiNormRelL2GetBufferHostSize\_16u\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_16u\_C4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.106.2.35 NppStatus nppiNormRelL2GetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_AC4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

### 7.106.2.36 NppStatus nppiNormRelL2GetBufferHostSize\_32f\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_C1MR.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.37 NppStatus nppiNormRelL2GetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_32f\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.38 NppStatus nppiNormRelL2GetBufferHostSize\_32f\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_32f\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.39 NppStatus nppiNormRelL2GetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_32f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.40 NppStatus nppiNormRelL2GetBufferHostSize\_32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.41 NppStatus nppiNormRelL2GetBufferHostSize\_8s\_C1MR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8s\_C1MR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.42 NppStatus nppiNormRelL2GetBufferHostSize\_8s\_C3CMR (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8s\_C3CMR.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.43 NppStatus nppiNormRelL2GetBufferHostSize\_8u\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.44 NppStatus nppiNormRelL2GetBufferHostSize\_8u\_C1MR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_8u\_C1MR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.45 NppStatus nppiNormRelL2GetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_8u\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.46 NppStatus nppiNormRelL2GetBufferHostSize\_8u\_C3CMR (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for *nppiNormRel\_L2\_8u\_C3CMR*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.47 NppStatus nppiNormRelL2GetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.106.2.48 NppStatus nppiNormRelL2GetBufferHostSize\_8u\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Computes the device scratch buffer size (in bytes) for nppiNormRel\_L2\_8u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.107 DotProd

Primitives for computing the dot product of two images.

### DotProd

Given two images  $pSrc1$  and  $pSrc2$  both with width  $W$  and height  $H$ , the dot product will be computed as

$$DotProd = \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [pSrc1(j, i) \cdot pSrc2(j, i)]$$

The functions require additional scratch buffer for computations.

- **NppStatus nppiDotProd\_8u64f\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image DotProd.*
- **NppStatus nppiDotProd\_8s64f\_C1R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit signed image DotProd.*
- **NppStatus nppiDotProd\_16u64f\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image DotProd.*
- **NppStatus nppiDotProd\_16s64f\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image DotProd.*
- **NppStatus nppiDotProd\_32u64f\_C1R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit unsigned image DotProd.*
- **NppStatus nppiDotProd\_32s64f\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed image DotProd.*
- **NppStatus nppiDotProd\_32f64f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pDp, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image DotProd.*
- **NppStatus nppiDotProd\_8u64f\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image DotProd.*
- **NppStatus nppiDotProd\_8s64f\_C3R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit signed image DotProd.*
- **NppStatus nppiDotProd\_16u64f\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)

*Three-channel 16-bit unsigned image DotProd.*

- `NppStatus nppiDotProd_16s64f_C3R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` \*pDeviceBuffer)

*Three-channel 16-bit signed image DotProd.*

- `NppStatus nppiDotProd_32u64f_C3R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` \*pDeviceBuffer)

*Three-channel 32-bit unsigned image DotProd.*

- `NppStatus nppiDotProd_32s64f_C3R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` \*pDeviceBuffer)

*Three-channel 32-bit signed image DotProd.*

- `NppStatus nppiDotProd_32f64f_C3R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` \*pDeviceBuffer)

*Three-channel 32-bit floating point image DotProd.*

- `NppStatus nppiDotProd_8u64f_C4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image DotProd.*

- `NppStatus nppiDotProd_8s64f_C4R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit signed image DotProd.*

- `NppStatus nppiDotProd_16u64f_C4R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit unsigned image DotProd.*

- `NppStatus nppiDotProd_16s64f_C4R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit signed image DotProd.*

- `NppStatus nppiDotProd_32u64f_C4R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit unsigned image DotProd.*

- `NppStatus nppiDotProd_32s64f_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit signed image DotProd.*

- `NppStatus nppiDotProd_32f64f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[4], `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image DotProd.*

- `NppStatus nppiDotProd_8u64f_AC4R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` aDp[3], `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image DotProd ignoring alpha channel.*

- **NppStatus** **nppiDotProd\_8s64f\_AC4R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit signed image DotProd ignoring alpha channel.*
- **NppStatus** **nppiDotProd\_16u64f\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image DotProd ignoring alpha channel.*
- **NppStatus** **nppiDotProd\_16s64f\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit signed image DotProd ignoring alpha channel.*
- **NppStatus** **nppiDotProd\_32u64f\_AC4R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 32-bit unsigned image DotProd ignoring alpha channel.*
- **NppStatus** **nppiDotProd\_32s64f\_AC4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 32-bit signed image DotProd ignoring alpha channel.*
- **NppStatus** **nppiDotProd\_32f64f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** aDp[3], **Npp8u** \*pDeviceBuffer)  
*Four-channel 32-bit floating point image DotProd ignoring alpha channel.*

## DotProdGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the Mean\_StdDev primitives.

- **NppStatus** **nppiDotProdGetBufferHostSize\_8u64f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8u64f\_C1R.*
- **NppStatus** **nppiDotProdGetBufferHostSize\_8s64f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8s64f\_C1R.*
- **NppStatus** **nppiDotProdGetBufferHostSize\_16u64f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_16u64f\_C1R.*
- **NppStatus** **nppiDotProdGetBufferHostSize\_16s64f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_16s64f\_C1R.*
- **NppStatus** **nppiDotProdGetBufferHostSize\_32u64f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32u64f\_C1R.*
- **NppStatus** **nppiDotProdGetBufferHostSize\_32s64f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_C1R.*
- **NppStatus** **nppiDotProdGetBufferHostSize\_32f64f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32f64f\_C1R.*

- [NppStatus nppiDotProdGetBufferHostSize\\_8u64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8u64f\_C3R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_8s64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8s64f\_C3R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_16u64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_16u64f\_C3R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_16s64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_16s64f\_C3R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_32u64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32u64f\_C3R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_32s64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_C3R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_32f64f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32f64f\_C3R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_8u64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8u64f\_C4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_8s64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8s64f\_C4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_16u64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_16u64f\_C4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_16s64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_16s64f\_C4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_32u64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32u64f\_C4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_32s64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_C4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_32f64f\\_C4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_32f64f\_C4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_8u64f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8u64f\_AC4R.*
- [NppStatus nppiDotProdGetBufferHostSize\\_8s64f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiDotProd\_8s64f\_AC4R.*



- [NppStatus](#)    [nppiDotProdGetBufferHostSize\\_16u64f\\_AC4R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#) \*[hpBufferSize](#))  
*Device scratch buffer size (in bytes) for nppiDotProd\_16u64f\_AC4R.*
- [NppStatus](#)    [nppiDotProdGetBufferHostSize\\_16s64f\\_AC4R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#) \*[hpBufferSize](#))  
*Device scratch buffer size (in bytes) for nppiDotProd\_16s64f\_AC4R.*
- [NppStatus](#)    [nppiDotProdGetBufferHostSize\\_32u64f\\_AC4R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#) \*[hpBufferSize](#))  
*Device scratch buffer size (in bytes) for nppiDotProd\_32u64f\_AC4R.*
- [NppStatus](#)    [nppiDotProdGetBufferHostSize\\_32s64f\\_AC4R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#) \*[hpBufferSize](#))  
*Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_AC4R.*
- [NppStatus](#)    [nppiDotProdGetBufferHostSize\\_32f64f\\_AC4R](#)    ([NppiSize](#)    [oSizeROI](#),    [int](#) \*[hpBufferSize](#))  
*Device scratch buffer size (in bytes) for nppiDotProd\_32f64f\_AC4R.*

### 7.107.1 Detailed Description

Primitives for computing the dot product of two images.

### 7.107.2 Function Documentation

#### 7.107.2.1 [NppStatus](#) [nppiDotProd\\_16s64f\\_AC4R](#) ([const](#) [Npp16s](#) \* [pSrc1](#), [int](#) [nSrc1Step](#), [const](#) [Npp16s](#) \* [pSrc2](#), [int](#) [nSrc2Step](#), [NppiSize](#) [oSizeROI](#), [Npp64f](#) [aDp](#)[3], [Npp8u](#) \* [pDeviceBuffer](#))

Four-channel 16-bit signed image DotProd ignoring alpha channel.

#### Parameters:

- [pSrc1](#) [Source-Image Pointer](#).
- [nSrc1Step](#) [Source-Image Line Step](#).
- [pSrc2](#) [Source-Image Pointer](#).
- [nSrc2Step](#) [Source-Image Line Step](#).
- [oSizeROI](#) [Region-of-Interest \(ROI\)](#).
- [aDp](#) Array that contains the computed dot product of the two images.
- [pDeviceBuffer](#) [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
 Use [nppiDotProdGetBufferHostSize\\_16s64f\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.2** `NppStatus nppiDotProd_16s64f_C1R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDp* Pointer to the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_16s64f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.3** `NppStatus nppiDotProd_16s64f_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_16s64f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.4** `NppStatus nppiDotProd_16s64f_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aDp* Array that contains the computed dot product of the two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiDotProdGetBufferHostSize\\_16s64f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.5** `NppStatus nppiDotProd_16u64f_AC4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image DotProd ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aDp* Array that contains the computed Inf-norm of differences.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiDotProdGetBufferHostSize\\_16u64f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.6** `NppStatus nppiDotProd_16u64f_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDp* Pointer to the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_16u64f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.7** `NppStatus nppiDotProd_16u64f_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_16u64f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.8** `NppStatus nppiDotProd_16u64f_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed Inf-norm of differences.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_16u64f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.9** `NppStatus nppiDotProd_32f64f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image DotProd ignoring alpha channel.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_32f64f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.10** `NppStatus nppiDotProd_32f64f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image DotProd.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDp* Pointer to the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_32f64f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.11** `NppStatus nppiDotProd_32f64f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aDp* Array that contains the computed dot product of the two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiDotProdGetBufferHostSize\\_32f64f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.12** `NppStatus nppiDotProd_32f64f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aDp* Array that contains the computed dot product of the two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiDotProdGetBufferHostSize\\_32f64f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.13** `NppStatus nppiDotProd_32s64f_AC4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image DotProd ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_32s64f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.14** `NppStatus nppiDotProd_32s64f_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDp* Pointer to the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_32s64f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.15** `NppStatus nppiDotProd_32s64f_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_32s64f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.16** `NppStatus nppiDotProd_32s64f_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_32s64f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.17** `NppStatus nppiDotProd_32u64f_AC4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image DotProd ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_32u64f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.18** `NppStatus nppiDotProd_32u64f_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image DotProd.



**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pDp* Pointer to the computed dot product of the two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiDotProdGetBufferHostSize\\_32u64f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.19** `NppStatus nppiDotProd_32u64f_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aDp* Array that contains the computed dot product of the two images.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiDotProdGetBufferHostSize\\_32u64f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.20** `NppStatus nppiDotProd_32u64f_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferSize\\_32u64f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.21** `NppStatus nppiDotProd_8s64f_AC4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image DotProd ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferSize\\_8s64f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.22** `NppStatus nppiDotProd_8s64f_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pDp* Pointer to the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferSize\\_8s64f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.23** `NppStatus nppiDotProd_8s64f_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_8s64f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.24** `NppStatus nppiDotProd_8s64f_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_8s64f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.25** `NppStatus nppiDotProd_8u64f_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image DotProd ignoring alpha channel.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_8u64f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.26** `NppStatus nppiDotProd_8u64f_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pDp, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image DotProd.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pDp* Pointer to the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_8u64f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.27** `NppStatus nppiDotProd_8u64f_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[3], Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image DotProd.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_8u64f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.28** `NppStatus nppiDotProd_8u64f_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f aDp[4], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image DotProd.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aDp* Array that contains the computed dot product of the two images.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiDotProdGetBufferHostSize\\_8u64f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.107.2.29** `NppStatus nppiDotProdGetBufferHostSize_16s64f_AC4R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiDotProd\_16s64f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.30** `NppStatus nppiDotProdGetBufferHostSize_16s64f_C1R (NppiSize oSizeROI, int * hpBufferSize)`

Device scratch buffer size (in bytes) for nppiDotProd\_16s64f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.31 NppStatus nppiDotProdGetBufferHostSize\_16s64f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_16s64f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.32 NppStatus nppiDotProdGetBufferHostSize\_16s64f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_16s64f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.33 NppStatus nppiDotProdGetBufferHostSize\_16u64f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_16u64f\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.34 NppStatus nppiDotProdGetBufferHostSize\_16u64f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_16u64f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.35 NppStatus nppiDotProdGetBufferHostSize\_16u64f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_16u64f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.36 NppStatus nppiDotProdGetBufferHostSize\_16u64f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_16u64f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.37 NppStatus nppiDotProdGetBufferHostSize\_32f64f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_32f64f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.38 NppStatus nppiDotProdGetBufferHostSize\_32f64f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_32f64f\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.39 NppStatus nppiDotProdGetBufferHostSize\_32f64f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_32f64f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.40 NppStatus nppiDotProdGetBufferHostSize\_32f64f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_32f64f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.107.2.41 NppStatus nppiDotProdGetBufferHostSize\_32s64f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.42 NppStatus nppiDotProdGetBufferHostSize\_32s64f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.43 NppStatus nppiDotProdGetBufferHostSize\_32s64f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.44 NppStatus nppiDotProdGetBufferHostSize\_32s64f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiDotProd\_32s64f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.45 NppStatus nppiDotProdGetBufferHostSize\_32u64f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiDotProd\_32u64f\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.46 NppStatus nppiDotProdGetBufferHostSize\_32u64f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiDotProd\_32u64f\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.47 NppStatus nppiDotProdGetBufferHostSize\_32u64f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for *nppiDotProd\_32u64f\_C3R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.48 NppStatus nppiDotProdGetBufferHostSize\_32u64f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_32u64f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.49 NppStatus nppiDotProdGetBufferHostSize\_8s64f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_8s64f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.50 NppStatus nppiDotProdGetBufferHostSize\_8s64f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_8s64f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.51 NppStatus nppiDotProdGetBufferHostSize\_8s64f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiDotProd\_8s64f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.52 NppStatus nppiDotProdGetBufferHostSize\_8s64f\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_8s64f\_C4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.53 NppStatus nppiDotProdGetBufferHostSize\_8u64f\_AC4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_8u64f\_AC4R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.54 NppStatus nppiDotProdGetBufferHostSize\_8u64f\_C1R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for *nppiDotProd\_8u64f\_C1R*.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.55 NppStatus nppiDotProdGetBufferHostSize\_8u64f\_C3R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiDotProd\_8u64f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.107.2.56 NppStatus nppiDotProdGetBufferHostSize\_8u64f\_C4R (NppiSize *oSizeROI*, int \*  
*hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiDotProd\_8u64f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.108 CountInRange.

Primitives for computing the amount of pixels that fall into the specified intensity range.

### CountInRange

The lower bound and the upper bound are inclusive.

The functions require additional scratch buffer for computations.

- `NppStatus nppiCountInRange_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int \*pCounts, `Npp8u` nLowerBound, `Npp8u` nUpperBound, `Npp8u` \*pDeviceBuffer)  
*One-channel 8-bit unsigned image CountInRange.*
- `NppStatus nppiCountInRange_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int \*pCounts, `Npp32f` nLowerBound, `Npp32f` nUpperBound, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit floating point image CountInRange.*
- `NppStatus nppiCountInRange_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp8u` aLowerBound[3], `Npp8u` aUpperBound[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image CountInRange.*
- `NppStatus nppiCountInRange_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp32f` aLowerBound[3], `Npp32f` aUpperBound[3], `Npp8u` \*pDeviceBuffer)  
*Three-channel 32-bit floating point image CountInRange.*
- `NppStatus nppiCountInRange_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp8u` aLowerBound[3], `Npp8u` aUpperBound[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image CountInRange ignoring alpha channel.*
- `NppStatus nppiCountInRange_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, int aCounts[3], `Npp32f` aLowerBound[3], `Npp32f` aUpperBound[3], `Npp8u` \*pDeviceBuffer)  
*Four-channel 32-bit floating point image CountInRange ignoring alpha channel.*

### CountInRangeGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CountInRange primitives.

- `NppStatus nppiCountInRangeGetBufferHostSize_8u_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiCountInRange\_8u\_C1R.*
- `NppStatus nppiCountInRangeGetBufferHostSize_32f_C1R` (`NppiSize` oSizeROI, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppiCountInRange\_32f\_C1R.*
- `NppStatus nppiCountInRangeGetBufferHostSize_8u_C3R` (`NppiSize` oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiCountInRange\_8u\_C3R.*

- [NppStatus](#) [nppiCountInRangeGetBufferHostSize\\_32f\\_C3R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiCountInRange\_32f\_C3R.*

- [NppStatus](#) [nppiCountInRangeGetBufferHostSize\\_8u\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiCountInRange\_8u\_AC4R.*

- [NppStatus](#) [nppiCountInRangeGetBufferHostSize\\_32f\\_AC4R](#) ([NppiSize](#) oSizeROI, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppiCountInRange\_32f\_AC4R.*

### 7.108.1 Detailed Description

Primitives for computing the amount of pixels that fall into the specified intensity range.

### 7.108.2 Function Documentation

**7.108.2.1** [NppStatus nppiCountInRange\\_32f\\_AC4R](#) (const [Npp32f](#) \* *pSrc*, int *nSrcStep*, [NppiSize](#) oSizeROI, int *aCounts*[3], [Npp32f](#) *aLowerBound*[3], [Npp32f](#) *aUpperBound*[3], [Npp8u](#) \* *pDeviceBuffer*)

Four-channel 32-bit floating point image CountInRange ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*aCounts* Array that contains the number of pixels that fall into the specified range for Three-channels.

*aLowerBound* Fixed size array of the lower bound of the specified range, one per channel.

*aUpperBound* Fixed size array of the upper bound of the specified range, one per channel.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiCountInRangeGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_RANGE\_ERROR if the lower bound is larger than the upper bound.

**7.108.2.2** [NppStatus nppiCountInRange\\_32f\\_C1R](#) (const [Npp32f](#) \* *pSrc*, int *nSrcStep*, [NppiSize](#) oSizeROI, int \* *pCounts*, [Npp32f](#) *nLowerBound*, [Npp32f](#) *nUpperBound*, [Npp8u](#) \* *pDeviceBuffer*)

One-channel 32-bit floating point image CountInRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pCounts* Pointer to the number of pixels that fall into the specified range.  
*nLowerBound* Lower bound of the specified range.  
*nUpperBound* Upper bound of the specified range.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiCountInRangeGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_RANGE\_ERROR if the lower bound is larger than the upper bound.

**7.108.2.3** `NppStatus nppiCountInRange_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp32f aLowerBound[3], Npp32f aUpperBound[3], Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CountInRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*aCounts* Array that contains the number of pixels that fall into the specified range for Three-channels.  
*aLowerBound* Fixed size array of the lower bound of the specified range, one per channel.  
*aUpperBound* Fixed size array of the upper bound of the specified range, one per channel.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiCountInRangeGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_RANGE\_ERROR if the lower bound is larger than the upper bound.

**7.108.2.4** `NppStatus nppiCountInRange_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp8u aLowerBound[3], Npp8u aUpperBound[3], Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CountInRange ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).



**aCounts** Array that contains the number of pixels that fall into the specified range for Three-channels.

**aLowerBound** Fixed size array of the lower bound of the specified range, one per channel.

**aUpperBound** Fixed size array of the upper bound of the specified range, one per channel.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiCountInRangeGetBufferHostSize\\_8u\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_RANGE\_ERROR if the lower bound is larger than the upper bound.

**7.108.2.5** NppStatus nppiCountInRange\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, int \* pCounts, Npp8u nLowerBound, Npp8u nUpperBound, Npp8u \* pDeviceBuffer)

One-channel 8-bit unsigned image CountInRange.

#### Parameters:

**pSrc** [Source-Image Pointer](#).

**nSrcStep** [Source-Image Line Step](#).

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**pCounts** Pointer to the number of pixels that fall into the specified range.

**nLowerBound** Lower bound of the specified range.

**nUpperBound** Upper bound of the specified range.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiCountInRangeGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_RANGE\_ERROR if the lower bound is larger than the upper bound.

**7.108.2.6** NppStatus nppiCountInRange\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, NppiSize oSizeROI, int aCounts[3], Npp8u aLowerBound[3], Npp8u aUpperBound[3], Npp8u \* pDeviceBuffer)

Three-channel 8-bit unsigned image CountInRange.

#### Parameters:

**pSrc** [Source-Image Pointer](#).

**nSrcStep** [Source-Image Line Step](#).

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**aCounts** Array that contains the number of pixels that fall into the specified range for Three-channels.

**aLowerBound** Fixed size array of the lower bound of the specified range, one per channel.

**aUpperBound** Fixed size array of the upper bound of the specified range, one per channel.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiCountInRangeGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_RANGE\_ERROR if the lower bound is larger than the upper bound.

**7.108.2.7 NppStatus nppiCountInRangeGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiCountInRange\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.108.2.8 NppStatus nppiCountInRangeGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiCountInRange\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.108.2.9 NppStatus nppiCountInRangeGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppiCountInRange\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.108.2.10 NppStatus nppiCountInRangeGetBufferHostSize\_8u\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiCountInRange\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.108.2.11 NppStatus nppiCountInRangeGetBufferHostSize\_8u\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiCountInRange\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.108.2.12 NppStatus nppiCountInRangeGetBufferHostSize\_8u\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppiCountInRange\_8u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.109 MaxEvery

Primitives for computing the maximal value of the pixel pair from two images.

### MaxEvery

The maximum is stored into the second image.

- [NppStatus nppiMaxEvery\\_8u\\_C1IR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 8-bit unsigned image MaxEvery.*
- [NppStatus nppiMaxEvery\\_16u\\_C1IR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 16-bit unsigned image MaxEvery.*
- [NppStatus nppiMaxEvery\\_16s\\_C1IR](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 16-bit signed image MaxEvery.*
- [NppStatus nppiMaxEvery\\_32f\\_C1IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 32-bit floating point image MaxEvery.*
- [NppStatus nppiMaxEvery\\_8u\\_C3IR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 8-bit unsigned image MaxEvery.*
- [NppStatus nppiMaxEvery\\_16u\\_C3IR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 16-bit unsigned image MaxEvery.*
- [NppStatus nppiMaxEvery\\_16s\\_C3IR](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 16-bit signed image MaxEvery.*
- [NppStatus nppiMaxEvery\\_32f\\_C3IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 32-bit floating point image MaxEvery.*
- [NppStatus nppiMaxEvery\\_8u\\_C4IR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Four-channel 8-bit unsigned image MaxEvery.*
- [NppStatus nppiMaxEvery\\_16u\\_C4IR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Four-channel 16-bit unsigned image MaxEvery.*
- [NppStatus nppiMaxEvery\\_16s\\_C4IR](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Four-channel 16-bit signed image MaxEvery.*

- **NppStatus nppiMaxEvery\_32f\_C4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 32-bit floating point image MaxEvery.*

- **NppStatus nppiMaxEvery\_8u\_AC4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 8-bit unsigned image MaxEvery ignoring alpha channel.*

- **NppStatus nppiMaxEvery\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 16-bit unsigned image MaxEvery ignoring alpha channel.*

- **NppStatus nppiMaxEvery\_16s\_AC4IR** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 16-bit signed image MaxEvery ignoring alpha channel.*

- **NppStatus nppiMaxEvery\_32f\_AC4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 32-bit floating point image MaxEvery ignoring alpha channel.*

## 7.109.1 Detailed Description

Primitives for computing the maximal value of the pixel pair from two images.

## 7.109.2 Function Documentation

### 7.109.2.1 **NppStatus nppiMaxEvery\_16s\_AC4IR** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit signed image MaxEvery ignoring alpha channel.

#### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pSrcDst** In-Place Image Pointer.

**nSrcDstStep** Source-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.109.2.2 `NppStatus nppiMaxEvery_16s_C1IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

One-channel 16-bit signed image MaxEvery.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.109.2.3 `NppStatus nppiMaxEvery_16s_C3IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Three-channel 16-bit signed image MaxEvery.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.109.2.4 `NppStatus nppiMaxEvery_16s_C4IR (const Npp16s * pSrc, int nSrcStep, Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI)`

Four-channel 16-bit signed image MaxEvery.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.5 NppStatus nppiMaxEvery\_16u\_AC4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four-channel 16-bit unsigned image MaxEvery ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.6 NppStatus nppiMaxEvery\_16u\_C1IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One-channel 16-bit unsigned image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.7 NppStatus nppiMaxEvery\_16u\_C3IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three-channel 16-bit unsigned image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.8 NppStatus nppiMaxEvery\_16u\_C4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four-channel 16-bit unsigned image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.9 NppStatus nppiMaxEvery\_32f\_AC4IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four-channel 32-bit floating point image MaxEvery ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.10 NppStatus nppiMaxEvery\_32f\_C1IR (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One-channel 32-bit floating point image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.



**7.109.2.11 NppStatus nppiMaxEvery\_32f\_C3IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit floating point image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.12 NppStatus nppiMaxEvery\_32f\_C4IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit floating point image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.13 NppStatus nppiMaxEvery\_8u\_AC4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit unsigned image MaxEvery ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.14 NppStatus nppiMaxEvery\_8u\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One-channel 8-bit unsigned image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.15 NppStatus nppiMaxEvery\_8u\_C3IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three-channel 8-bit unsigned image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.109.2.16 NppStatus nppiMaxEvery\_8u\_C4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit unsigned image MaxEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

## 7.110 MinEvery

Primitives for computing the minimal value of the pixel pair from two images.

### MinEvery

The minimum is stored into the second image.

- [NppStatus nppiMinEvery\\_8u\\_C1IR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 8-bit unsigned image MinEvery.*
- [NppStatus nppiMinEvery\\_16u\\_C1IR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 16-bit unsigned image MinEvery.*
- [NppStatus nppiMinEvery\\_16s\\_C1IR](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 16-bit signed image MinEvery.*
- [NppStatus nppiMinEvery\\_32f\\_C1IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*One-channel 32-bit floating point image MinEvery.*
- [NppStatus nppiMinEvery\\_8u\\_C3IR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 8-bit unsigned image MinEvery.*
- [NppStatus nppiMinEvery\\_16u\\_C3IR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 16-bit unsigned image MinEvery.*
- [NppStatus nppiMinEvery\\_16s\\_C3IR](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 16-bit signed image MinEvery.*
- [NppStatus nppiMinEvery\\_32f\\_C3IR](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Three-channel 32-bit floating point image MinEvery.*
- [NppStatus nppiMinEvery\\_8u\\_C4IR](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Four-channel 8-bit unsigned image MinEvery.*
- [NppStatus nppiMinEvery\\_16u\\_C4IR](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)  
*Four-channel 16-bit unsigned image MinEvery.*
- [NppStatus nppiMinEvery\\_16s\\_C4IR](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI)

*Four-channel 16-bit signed image MinEvery.*

- **NppStatus nppiMinEvery\_32f\_C4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 32-bit floating point image MinEvery.*

- **NppStatus nppiMinEvery\_8u\_AC4IR** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 8-bit unsigned image MinEvery ignoring alpha channel.*

- **NppStatus nppiMinEvery\_16u\_AC4IR** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 16-bit unsigned image MinEvery ignoring alpha channel.*

- **NppStatus nppiMinEvery\_16s\_AC4IR** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 16-bit signed image MinEvery ignoring alpha channel.*

- **NppStatus nppiMinEvery\_32f\_AC4IR** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

*Four-channel 32-bit floating point image MinEvery ignoring alpha channel.*

### 7.110.1 Detailed Description

Primitives for computing the minimal value of the pixel pair from two images.

### 7.110.2 Function Documentation

#### 7.110.2.1 **NppStatus nppiMinEvery\_16s\_AC4IR** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI)

Four-channel 16-bit signed image MinEvery ignoring alpha channel.

##### Parameters:

**pSrc** Source-Image Pointer.

**nSrcStep** Source-Image Line Step.

**pSrcDst** In-Place Image Pointer.

**nSrcDstStep** Source-Image Line Step.

**oSizeROI** Region-of-Interest (ROI).

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.2 NppStatus nppiMinEvery\_16s\_C1IR (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One-channel 16-bit signed image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.3 NppStatus nppiMinEvery\_16s\_C3IR (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three-channel 16-bit signed image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.4 NppStatus nppiMinEvery\_16s\_C4IR (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 16-bit signed image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.5 NppStatus nppiMinEvery\_16u\_AC4IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Four-channel 16-bit unsigned image MinEvery ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.6 NppStatus nppiMinEvery\_16u\_C1IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

One-channel 16-bit unsigned image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.7 NppStatus nppiMinEvery\_16u\_C3IR (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI)**

Three-channel 16-bit unsigned image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.8 NppStatus nppiMinEvery\_16u\_C4IR (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 16-bit unsigned image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.9 NppStatus nppiMinEvery\_32f\_AC4IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit floating point image MinEvery ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.10 NppStatus nppiMinEvery\_32f\_C1IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One-channel 32-bit floating point image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.11 NppStatus nppiMinEvery\_32f\_C3IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three-channel 32-bit floating point image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.12 NppStatus nppiMinEvery\_32f\_C4IR (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 32-bit floating point image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.13 NppStatus nppiMinEvery\_8u\_AC4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit unsigned image MinEvery ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.



**7.110.2.14 NppStatus nppiMinEvery\_8u\_C1IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

One-channel 8-bit unsigned image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.15 NppStatus nppiMinEvery\_8u\_C3IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Three-channel 8-bit unsigned image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.110.2.16 NppStatus nppiMinEvery\_8u\_C4IR (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*)**

Four-channel 8-bit unsigned image MinEvery.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

## 7.111 Integral

Primitives for computing the integral image of a given image.

### Integral

Given an input image  $pSrc$  and the specified value  $nVal$ , the pixel value of the integral image  $pDst$  at coordinate  $(i, j)$  will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

If the size of the input image is  $W \times H$ , the size of the integral image will be  $(W + 1) \times (H + 1)$ .

- [NppStatus nppiIntegral\\_8u32s\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp32s](#) \*pDst, int nDstStep, [NppiSize](#) oROI, [Npp32s](#) nVal)

*One-channel 8-bit unsigned image Integral with 32-bit signed output.*

- [NppStatus nppiIntegral\\_8u32f\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oROI, [Npp32f](#) nVal)

*One-channel 8-bit unsigned image Integral with 32-bit floating point output.*

### 7.111.1 Detailed Description

Primitives for computing the integral image of a given image.

### 7.111.2 Function Documentation

#### 7.111.2.1 [NppStatus nppiIntegral\\_8u32f\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oROI, [Npp32f](#) nVal)

One-channel 8-bit unsigned image Integral with 32-bit floating point output.

#### Parameters:

- [pSrc](#) [Source-Image Pointer](#).
- [nSrcStep](#) [Source-Image Line Step](#).
- [pDst](#) [Destination-Image Pointer](#).
- [nDstStep](#) [Destination-Image Line Step](#).
- [oROI](#) [Region-of-Interest \(ROI\)](#).
- [nVal](#) The value to add to pDst image pixels

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.111.2.2** `NppStatus nppiIntegral_8u32s_C1R (const Npp8u * pSrc, int nSrcStep, Npp32s * pDst, int nDstStep, NppiSize oROI, Npp32s nVal)`

One-channel 8-bit unsigned image Integral with 32-bit signed output.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oROI* Region-of-Interest (ROI).

*nVal* The value to add to pDst image pixels

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

## 7.112 SqrIntegral

Primitives for computing both the integral and the squared integral images of a given image.

### SqrIntegral

Given an input image  $pSrc$  and the specified value  $nVal$ , the pixel value of the integral image  $pDst$  at coordinate  $(i, j)$  will be computed as

$$pDst(j, i) = nVal + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)$$

Given an input image  $pSrc$  and the specified value  $nValSqr$ , the pixel value of the squared integral image  $pSqr$  at coordinate  $(i, j)$  will be computed as

$$pSqr(j, i) = nValSqr + \sum_{l=0}^{j-1} \sum_{k=0}^{i-1} pSrc(l, k)^2$$

If the size of the input image is  $W \times H$ , the size of the squared integral image will be  $(W + 1) \times (H + 1)$ .

- **NppStatus nppiSqrIntegral\_8u32s\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **Npp32s** \*pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32s** nVal, **Npp32s** nValSqr)  
*One-channel 8-bit unsigned image SqrIntegral.*
- **NppStatus nppiSqrIntegral\_8u32s64f\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32s** \*pDst, int nDstStep, **Npp64f** \*pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32s** nVal, **Npp64f** nValSqr)  
*One-channel 8-bit unsigned image SqrIntegral.*
- **NppStatus nppiSqrIntegral\_8u32f64f\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **Npp64f** \*pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32f** nVal, **Npp64f** nValSqr)  
*One-channel 8-bit unsigned image SqrIntegral.*

### 7.112.1 Detailed Description

Primitives for computing both the integral and the squared integral images of a given image.

### 7.112.2 Function Documentation

**7.112.2.1 NppStatus nppiSqrIntegral\_8u32f64f\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **Npp64f** \*pSqr, int nSqrStep, **NppiSize** oSrcROI, **Npp32f** nVal, **Npp64f** nValSqr)

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit floating point. Destination square integral image is 64-bit double floating point.

**Parameters:**

**pSrc** Source-Image Pointer.

*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pSqr* Destination-Image Pointer.  
*nSqrStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).  
*nVal* The value to add to pDst image pixels  
*nValSqr* The value to add to pSqr image pixels

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.112.2.2** `NppStatus nppiSqrIntegral_8u32s64f_C1R (const Npp8u *pSrc, int nSrcStep, Npp32s *pDst, int nDstStep, Npp64f *pSqr, int nSqrStep, NppiSize oSrcROI, Npp32s nVal, Npp64f nValSqr)`

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image is 32-bit signed int. Destination square integral image is 64-bit double floating point.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*pSqr* Destination-Image Pointer.  
*nSqrStep* Destination-Image Line Step.  
*oSrcROI* Region-of-Interest (ROI).  
*nVal* The value to add to pDst image pixels  
*nValSqr* The value to add to pSqr image pixels

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.112.2.3** `NppStatus nppiSqrIntegral_8u32s_C1R (const Npp8u *pSrc, int nSrcStep, Npp32s *pDst, int nDstStep, Npp32s *pSqr, int nSqrStep, NppiSize oSrcROI, Npp32s nVal, Npp32s nValSqr)`

One-channel 8-bit unsigned image SqrIntegral.

Destination integral image and square integral image are 32-bit signed int.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pSqr* Destination-Image Pointer.

*nSqrStep* Destination-Image Line Step.

*oSrcROI* Region-of-Interest (ROI).

*nVal* The value to add to pDst image pixels

*nValSqr* The value to add to pSqr image pixels

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.113 RectStdDev

Primitives for computing the standard deviation of the integral images.

### RectStdDev

- **NppStatus** **nppiRectStdDev\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp64f** \*pSqr, int nSqrStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect)  
*One-channel 32-bit floating point image RectStdDev.*
- **NppStatus** **nppiRectStdDev\_32s\_C1RSfs** (const **Npp32s** \*pSrc, int nSrcStep, const **Npp32s** \*pSqr, int nSqrStep, **Npp32s** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect, int nScaleFactor)  
*One-channel 32-bit signed image RectStdDev, scaled by  $2^l - nScaleFactor$ .*
- **NppStatus** **nppiRectStdDev\_32s32f\_C1R** (const **Npp32s** \*pSrc, int nSrcStep, const **Npp64f** \*pSqr, int nSqrStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppiRect** oRect)  
*One-channel 32-bit signed image RectStdDev.*

### 7.113.1 Detailed Description

Primitives for computing the standard deviation of the integral images.

The function computes the standard deviation of the pixel in the rectangular window with the integral image *pSrc* and the squared integral image *pSqr*, which can be obtained by calling [Integral](#) and [SqrIntegral](#).

The standard deviation of the pixel  $(j, i)$  can be computed using the formula:

$$pDst(j, i) = \sqrt{\max(0, \frac{\sum(SqrIntegral) \cdot N - (\sum(Integral))^2}{N^2})}$$

where  $\sum(SqrIntegral) = pSqr[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSqr[j + oRect.y, i + oRect.x + oRect.width] - pSqr[j + oRect.y + oRect.height, i + oRect.x] + pSqr[j + oRect.y, i + oRect.x]$ ,  $\sum(Integral) = pSrc[j + oRect.y + oRect.height, i + oRect.x + oRect.width] - pSrc[j + oRect.y, i + oRect.x + oRect.width] - pSrc[j + oRect.y + oRect.height, i + oRect.x] + pSrc[j + oRect.y, i + oRect.x]$ ,  $N = oRect.width \cdot oRect.height$ .

The size of the *pSrc* and *pSqr* should be  $(oSizeROI.width + oRect.x + oRect.width, oSizeROI.height + oRect.y + oRect.height)$ .

### 7.113.2 Function Documentation

#### 7.113.2.1 NppStatus nppiRectStdDev\_32f\_C1R (const Npp32f \*pSrc, int nSrcStep, const Npp64f \*pSqr, int nSqrStep, Npp32f \*pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect)

One-channel 32-bit floating point image RectStdDev.

#### Parameters:

- pSrc** [Source-Image Pointer](#).
- nSrcStep** [Source-Image Line Step](#).
- pSqr** [Destination-Image Pointer](#).

*nSqrStep* Destination-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oRect* rectangular window

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.113.2.2** `NppStatus nppiRectStdDev_32s32f_C1R (const Npp32s * pSrc, int nSrcStep, const Npp64f * pSqr, int nSqrStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect)`

One-channel 32-bit signed image RectStdDev.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSqr* Destination-Image Pointer.

*nSqrStep* Destination-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oRect* rectangular window

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.113.2.3** `NppStatus nppiRectStdDev_32s_C1RSfs (const Npp32s * pSrc, int nSrcStep, const Npp32s * pSqr, int nSqrStep, Npp32s * pDst, int nDstStep, NppiSize oSizeROI, NppiRect oRect, int nScaleFactor)`

One-channel 32-bit signed image RectStdDev, scaled by  $2^{(-nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pSqr* Destination-Image Pointer.

*nSqrStep* Destination-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*oRect* rectangular window



*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.114 HistogramEven

Primitives for computing the histogram of an image with evenly distributed bins.

### HistogramEven

The *nLowerLevel* (inclusive) and *nUpperLevel* (exclusive) define the boundaries of the range, which are evenly segmented into  $nLevel - 1$  bins.

The computed histogram is stored in *pHist*. The levels are calculated by another primitive [nppiEvenLevelsHost\\_32s](#) and are stored in a host pointer *hpLevels*. The number of levels is also  $nLevel - 1$ . The histogram  $pHist[k]$  is defined as the total number of pixels that fall into the range:  $hpLevels[k] \leq pSrc(j, i) < hpLevels[k + 1]$ . The functions require additional scratch buffer for computations.

- [NppStatus nppiEvenLevelsHost\\_32s](#) ([Npp32s](#) \*hpLevels, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel)  
*Compute levels with even distribution.*
- [NppStatus nppiHistogramEven\\_8u\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) \*pBuffer)  
*One-channel 8-bit unsigned HistogramEven.*
- [NppStatus nppiHistogramEven\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*Three-channel 8-bit unsigned HistogramEven.*
- [NppStatus nppiHistogramEven\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[4], int nLevels[4], [Npp32s](#) nLowerLevel[4], [Npp32s](#) nUpperLevel[4], [Npp8u](#) \*pBuffer)  
*Four-channel 8-bit unsigned HistogramEven.*
- [NppStatus nppiHistogramEven\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*Four-channel 8-bit unsigned HistogramEven ignoring alpha channel.*
- [NppStatus nppiHistogramEven\\_16u\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist, int nLevels, [Npp32s](#) nLowerLevel, [Npp32s](#) nUpperLevel, [Npp8u](#) \*pBuffer)  
*One-channel 16-bit unsigned HistogramEven.*
- [NppStatus nppiHistogramEven\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[3], int nLevels[3], [Npp32s](#) nLowerLevel[3], [Npp32s](#) nUpperLevel[3], [Npp8u](#) \*pBuffer)  
*Three-channel 16-bit unsigned HistogramEven.*
- [NppStatus nppiHistogramEven\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSizeROI, [Npp32s](#) \*pHist[4], int nLevels[4], [Npp32s](#) nLowerLevel[4], [Npp32s](#) nUpperLevel[4], [Npp8u](#) \*pBuffer)  
*Four-channel 16-bit unsigned HistogramEven.*

- `NppStatus nppiHistogramEven_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[3], int nLevels[3], `Npp32s` nLowerLevel[3], `Npp32s` nUpperLevel[3], `Npp8u` \*pBuffer)

*Four-channel 16-bit unsigned HistogramEven ignoring alpha channel.*

- `NppStatus nppiHistogramEven_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist, int nLevels, `Npp32s` nLowerLevel, `Npp32s` nUpperLevel, `Npp8u` \*pBuffer)

*One-channel 16-bit signed HistogramEven.*

- `NppStatus nppiHistogramEven_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[3], int nLevels[3], `Npp32s` nLowerLevel[3], `Npp32s` nUpperLevel[3], `Npp8u` \*pBuffer)

*Three-channel 16-bit signed HistogramEven.*

- `NppStatus nppiHistogramEven_16s_C4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[4], int nLevels[4], `Npp32s` nLowerLevel[4], `Npp32s` nUpperLevel[4], `Npp8u` \*pBuffer)

*Four-channel 16-bit signed HistogramEven.*

- `NppStatus nppiHistogramEven_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `NppiSize` oSizeROI, `Npp32s` \*pHist[3], int nLevels[3], `Npp32s` nLowerLevel[3], `Npp32s` nUpperLevel[3], `Npp8u` \*pBuffer)

*Four-channel 16-bit signed HistogramEven ignoring alpha channel.*

## HistogramEvenGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramEven primitives.

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C1R` (`NppiSize` oSizeROI, int nLevels, int \*hpBufferSize)

*Buffer size for `nppiHistogramEven_8u_C1R`.*

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C3R` (`NppiSize` oSizeROI, int nLevels[3], int \*hpBufferSize)

*Buffer size for `nppiHistogramEven_8u_C3R`.*

- `NppStatus nppiHistogramEvenGetBufferSize_8u_C4R` (`NppiSize` oSizeROI, int nLevels[4], int \*hpBufferSize)

*Buffer size for `nppiHistogramEven_8u_C4R`.*

- `NppStatus nppiHistogramEvenGetBufferSize_8u_AC4R` (`NppiSize` oSizeROI, int nLevels[3], int \*hpBufferSize)

*Buffer size for `nppiHistogramEven_8u_AC4R`.*

- `NppStatus nppiHistogramEvenGetBufferSize_16u_C1R` (`NppiSize` oSizeROI, int nLevels, int \*hpBufferSize)

*Buffer size for `nppiHistogramEven_16u_C1R`.*

- **NppStatus** `nppiHistogramEvenGetBufferSize_16u_C3R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Buffer size for `nppiHistogramEven_16u_C3R`.*
- **NppStatus** `nppiHistogramEvenGetBufferSize_16u_C4R` (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Buffer size for `nppiHistogramEven_16u_C4R`.*
- **NppStatus** `nppiHistogramEvenGetBufferSize_16u_AC4R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Buffer size for `nppiHistogramEven_16u_AC4R`.*
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_C1R` (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Buffer size for `nppiHistogramEven_16s_C1R`.*
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_C3R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Buffer size for `nppiHistogramEven_16s_C3R`.*
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_C4R` (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Buffer size for `nppiHistogramEven_16s_C4R`.*
- **NppStatus** `nppiHistogramEvenGetBufferSize_16s_AC4R` (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Buffer size for `nppiHistogramEven_16s_AC4R`.*

### 7.114.1 Detailed Description

Primitives for computing the histogram of an image with evenly distributed bins.

### 7.114.2 Function Documentation

#### 7.114.2.1 **NppStatus** `nppiEvenLevelsHost_32s` (**Npp32s** \* *hpLevels*, int *nLevels*, **Npp32s** *nLowerLevel*, **Npp32s** *nUpperLevel*)

Compute levels with even distribution.

#### Parameters:

- hpLevels*** A host pointer to array which receives the levels being computed. The array needs to be of size *nLevels*.
- nLevels*** The number of levels being computed. *nLevels* must be at least 2.
- nLowerLevel*** Lower boundary value of the lowest level.
- nUpperLevel*** Upper boundary value of the greatest level.

**Returns:**

image\_data\_error\_codes, or NPP\_HISTO\_NUMBER\_OF\_LEVELS\_ERROR if an invalid nLevels is specified.

**7.114.2.2** `NppStatus nppiHistogramEven_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

Four-channel 16-bit signed HistogramEven ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16s\\_AC4R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.114.2.3** `NppStatus nppiHistogramEven_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

One-channel 16-bit signed HistogramEven.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16s\\_C1R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.114.2.4 NppStatus nppiHistogramEven\_16s\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u \* *pBuffer*)

Three-channel 16-bit signed HistogramEven.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16s\\_C3R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.114.2.5 NppStatus nppiHistogramEven\_16s\_C4R** (const Npp16s \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], int *nLevels*[4], Npp32s *nLowerLevel*[4], Npp32s *nUpperLevel*[4], Npp8u \* *pBuffer*)

Four-channel 16-bit signed HistogramEven.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16s\\_C4R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.114.2.6** `NppStatus nppiHistogramEven_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

Four-channel 16-bit unsigned HistogramEven ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16u\\_AC4R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.114.2.7** `NppStatus nppiHistogramEven_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

One-channel 16-bit unsigned HistogramEven.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16u\\_C1R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.114.2.8 NppStatus nppiHistogramEven\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], int *nLevels*[3], Npp32s *nLowerLevel*[3], Npp32s *nUpperLevel*[3], Npp8u \* *pBuffer*)

Three-channel 16-bit unsigned HistogramEven.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16u\\_C3R](#)) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.114.2.9 NppStatus nppiHistogramEven\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], int *nLevels*[4], Npp32s *nLowerLevel*[4], Npp32s *nUpperLevel*[4], Npp8u \* *pBuffer*)

Four-channel 16-bit unsigned HistogramEven.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist*[*i*] be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_16u\\_C4R](#)) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.114.2.10** `NppStatus nppiHistogramEven_8u_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

Four-channel 8-bit unsigned HistogramEven ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by *pHist[i]* be of size *nLevels[i]*-1.

*nLevels* Array containing number of levels per color channel.

*nLowerLevel* Array containing lower-level of lowest bin per color channel.

*nUpperLevel* Array containing upper-level of highest bin per color channel.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_8u\\_AC4R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.114.2.11** `NppStatus nppiHistogramEven_8u_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, int nLevels, Npp32s nLowerLevel, Npp32s nUpperLevel, Npp8u * pBuffer)`

One-channel 8-bit unsigned HistogramEven.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*nLevels* Number of levels.

*nLowerLevel* Lower boundary of lowest level bin.

*nUpperLevel* Upper boundary of highest level bin.

*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_8u\\_C1R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.114.2.12** `NppStatus nppiHistogramEven_8u_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], int nLevels[3], Npp32s nLowerLevel[3], Npp32s nUpperLevel[3], Npp8u * pBuffer)`

Three-channel 8-bit unsigned HistogramEven.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.  
*nLevels* Array containing number of levels per color channel.  
*nLowerLevel* Array containing lower-level of lowest bin per color channel.  
*nUpperLevel* Array containing upper-level of highest bin per color channel.  
*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_8u\\_C3R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.114.2.13** `NppStatus nppiHistogramEven_8u_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], int nLevels[4], Npp32s nLowerLevel[4], Npp32s nUpperLevel[4], Npp8u * pBuffer)`

Four-channel 8-bit unsigned HistogramEven.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pHist* Array of pointers which are receiving computed histograms per color channel. Array pointed by pHist[i] be of size nLevels[i]-1.  
*nLevels* Array containing number of levels per color channel.  
*nLowerLevel* Array containing lower-level of lowest bin per color channel.  
*nUpperLevel* Array containing upper-level of highest bin per color channel.  
*pBuffer* Pointer to appropriately sized ([nppiHistogramEvenGetBufferSize\\_8u\\_C4R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.114.2.14** `NppStatus nppiHistogramEvenGetBufferSize_16s_AC4R (NppiSize oSizeROI, int nLevels[3], int * hpBufferSize)`

Buffer size for [nppiHistogramEven\\_16s\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nLevels* Array containing number of levels per color channel.  
*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.15 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Buffer size for [nppiHistogramEven\\_16s\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.16 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Buffer size for [nppiHistogramEven\\_16s\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.17 NppStatus nppiHistogramEvenGetBufferSize\_16s\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)**

Buffer size for [nppiHistogramEven\\_16s\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.18 NppStatus nppiHistogramEvenGetBufferSize\_16u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Buffer size for [nppiHistogramEven\\_16u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.19 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Buffer size for [nppiHistogramEven\\_16u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.20 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C3R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Buffer size for [nppiHistogramEven\\_16u\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.21 NppStatus nppiHistogramEvenGetBufferSize\_16u\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Buffer size for [nppiHistogramEven\\_16u\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.22 NppStatus nppiHistogramEvenGetBufferSize\_8u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Buffer size for [nppiHistogramEven\\_8u\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

**7.114.2.23 NppStatus nppiHistogramEvenGetBufferSize\_8u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Buffer size for [nppiHistogramEven\\_8u\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

#### 7.114.2.24 NppStatus nppiHistogramEvenGetBufferSize\_8u\_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)

Buffer size for [nppiHistogramEven\\_8u\\_C3R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

#### 7.114.2.25 NppStatus nppiHistogramEvenGetBufferSize\_8u\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)

Buffer size for [nppiHistogramEven\\_8u\\_C4R](#).

##### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

## 7.115 HistogramRange

Primitives for computing the histogram of an image within specified ranges.

### HistogramEven

The histogram is computed according to the ranges provided in *pLevels*.

The histogram  $pHist[k]$  is defined as the total number of pixels that fall into the range:  $pLevels[k] \leq pSrc(j, i) < pLevels[k + 1]$ . The number of the histogram bins is  $nLevel - 1$ . The functions require additional scratch buffer for computations.

- **NppStatus nppiHistogramRange\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32s** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*One-channel 8-bit unsigned HistogramRange.*
- **NppStatus nppiHistogramRange\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Three-channel 8-bit unsigned HistogramRange.*
- **NppStatus nppiHistogramRange\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32s** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*Four-channel 8-bit unsigned HistogramRange.*
- **NppStatus nppiHistogramRange\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Four-channel 8-bit unsigned HistogramRange ignoring alpha channel.*
- **NppStatus nppiHistogramRange\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32s** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*One-channel 16-bit unsigned HistogramRange.*
- **NppStatus nppiHistogramRange\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Three-channel 16-bit unsigned HistogramRange.*
- **NppStatus nppiHistogramRange\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32s** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*Four-channel 16-bit unsigned HistogramRange.*
- **NppStatus nppiHistogramRange\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Four-channel 16-bit unsigned HistogramRange ignoring alpha channel.*
- **NppStatus nppiHistogramRange\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32s** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*One-channel 16-bit signed HistogramRange.*
- **NppStatus nppiHistogramRange\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Three-channel 16-bit signed HistogramRange.*

- **NppStatus** **nppiHistogramRange\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32s** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*Four-channel 16-bit signed HistogramRange.*
- **NppStatus** **nppiHistogramRange\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32s** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Four-channel 16-bit signed HistogramRange.*
- **NppStatus** **nppiHistogramRange\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist, const **Npp32f** \*pLevels, int nLevels, **Npp8u** \*pBuffer)  
*One-channel 32-bit floating point HistogramRange.*
- **NppStatus** **nppiHistogramRange\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32f** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Three-channel 32-bit floating point HistogramRange.*
- **NppStatus** **nppiHistogramRange\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[4], const **Npp32f** \*pLevels[4], int nLevels[4], **Npp8u** \*pBuffer)  
*Four-channel 32-bit floating point HistogramRange.*
- **NppStatus** **nppiHistogramRange\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **NppiSize** oSizeROI, **Npp32s** \*pHist[3], const **Npp32f** \*pLevels[3], int nLevels[3], **Npp8u** \*pBuffer)  
*Four-channel 32-bit floating point HistogramRange ignoring alpha channel.*

## HistogramRangeGetBufferSize

Companion primitives for computing the device buffer size (in bytes) required by the HistogramRange primitives.

- **NppStatus** **nppiHistogramRangeGetBufferSize\_8u\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C1R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_8u\_C3R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C3R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_8u\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_C4R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_8u\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_8u\_AC4R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_16u\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C1R.*



- **NppStatus** **nppiHistogramRangeGetBufferSize\_16u\_C3R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C3R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_16u\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_C4R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_16u\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16u\_AC4R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_16s\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C1R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_16s\_C3R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C3R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_16s\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_C4R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_16s\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_16s\_AC4R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_32f\_C1R** (**NppiSize** oSizeROI, int nLevels, int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C1R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_32f\_C3R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C3R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_32f\_C4R** (**NppiSize** oSizeROI, int nLevels[4], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_32f\_C4R.*
- **NppStatus** **nppiHistogramRangeGetBufferSize\_32f\_AC4R** (**NppiSize** oSizeROI, int nLevels[3], int \*hpBufferSize)  
*Scratch-buffer size for nppiHistogramRange\_32f\_AC4R.*

### 7.115.1 Detailed Description

Primitives for computing the histogram of an image within specified ranges.

## 7.115.2 Function Documentation

**7.115.2.1** `NppStatus nppiHistogramRange_16s_AC4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 16-bit signed HistogramRange.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16s\\_AC4R](#)) scratch buffer.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.2** `NppStatus nppiHistogramRange_16s_C1R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

One-channel 16-bit signed HistogramRange.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16s\\_C1R](#)) scratch buffer.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.3** `NppStatus nppiHistogramRange_16s_C3R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Three-channel 16-bit signed HistogramRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16s\\_C3R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.4** `NppStatus nppiHistogramRange_16s_C4R (const Npp16s * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[4], const Npp32s * pLevels[4], int nLevels[4], Npp8u * pBuffer)`

Four-channel 16-bit signed HistogramRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16s\\_C4R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.5** `NppStatus nppiHistogramRange_16u_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Four-channel 16-bit unsigned HistogramRange ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16u\\_AC4R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.6** `NppStatus nppiHistogramRange_16u_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist, const Npp32s * pLevels, int nLevels, Npp8u * pBuffer)`

One-channel 16-bit unsigned HistogramRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size nLevels.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16u\\_C1R](#)) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.7** `NppStatus nppiHistogramRange_16u_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s * pHist[3], const Npp32s * pLevels[3], int nLevels[3], Npp8u * pBuffer)`

Three-channel 16-bit unsigned HistogramRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16u\\_C3R](#)) scratch buffer.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.8 NppStatus nppiHistogramRange\_16u\_C4R** (const Npp16u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4], Npp8u \* *pBuffer*)

Four-channel 16-bit unsigned HistogramRange.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_16u\\_C4R](#)) scratch buffer.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.9 NppStatus nppiHistogramRange\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], const Npp32f \* *pLevels*[3], int *nLevels*[3], Npp8u \* *pBuffer*)

Four-channel 32-bit floating point HistogramRange ignoring alpha channel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized (nppiHistogramRangeGetBufferSize\_32f\_AC4R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.10 NppStatus nppiHistogramRange\_32f\_C1R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist, const Npp32f \* pLevels, int nLevels, Npp8u \* pBuffer)**

One-channel 32-bit floating point HistogramRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size nLevels-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size nLevels.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized (nppiHistogramRangeGetBufferSize\_32f\_C1R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.11 NppStatus nppiHistogramRange\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, NppiSize oSizeROI, Npp32s \* pHist[3], const Npp32f \* pLevels[3], int nLevels[3], Npp8u \* pBuffer)**

Three-channel 32-bit floating point HistogramRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized (nppiHistogramRangeGetBufferSize\_32f\_C3R) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.12 NppStatus nppiHistogramRange\_32f\_C4R** (const Npp32f \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], const Npp32f \* *pLevels*[4], int *nLevels*[4], Npp8u \* *pBuffer*)

Four-channel 32-bit floating point HistogramRange.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_32f_C4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.115.2.13 NppStatus nppiHistogramRange\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3], Npp8u \* *pBuffer*)

Four-channel 8-bit unsigned HistogramRange ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized (`nppiHistogramRangeGetBufferSize_8u_AC4R`) scratch buffer.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.115.2.14 **NppStatus nppiHistogramRange\_8u\_C1R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*, const Npp32s \* *pLevels*, int *nLevels*, Npp8u \* *pBuffer*)

One-channel 8-bit unsigned HistogramRange.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Pointer to array that receives the computed histogram. The array must be of size *nLevels*-1.

*pLevels* Pointer to array containing the level sizes of the bins. The array must be of size *nLevels*.

*nLevels* Number of levels in histogram.

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_8u\\_C1R](#)) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.115.2.15 **NppStatus nppiHistogramRange\_8u\_C3R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[3], const Npp32s \* *pLevels*[3], int *nLevels*[3], Npp8u \* *pBuffer*)

Three-channel 8-bit unsigned HistogramRange.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by *pHist*[*i*] must be of size *nLevels*[*i*]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by *pLevel*[*i*] must be of size *nLevels*[*i*].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_8u\\_C3R](#)) scratch buffer.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.115.2.16 **NppStatus nppiHistogramRange\_8u\_C4R** (const Npp8u \* *pSrc*, int *nSrcStep*, NppiSize *oSizeROI*, Npp32s \* *pHist*[4], const Npp32s \* *pLevels*[4], int *nLevels*[4], Npp8u \* *pBuffer*)

Four-channel 8-bit unsigned HistogramRange.

##### Parameters:

*pSrc* [Source-Image Pointer](#).



*nSrcStep* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pHist* Array of pointers which are receiving the computed histograms per color channel. Array pointed by pHist[i] must be of size nLevels[i]-1.

*nLevels* Array containing number of levels per color channel.

*pLevels* Array containing pointers to level-arrays per color channel. Array pointed by pLevel[i] must be of size nLevels[i].

*pBuffer* Pointer to appropriately sized ([nppiHistogramRangeGetBufferSize\\_8u\\_C4R](#)) scratch buffer.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.115.2.17 NppStatus nppiHistogramRangeGetBufferSize\_16s\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)

Scratch-buffer size for nppiHistogramRange\_16s\_AC4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

#### 7.115.2.18 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)

Scratch-buffer size for nppiHistogramRange\_16s\_C1R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.19 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16s\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.20 NppStatus nppiHistogramRangeGetBufferSize\_16s\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16s\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.21 NppStatus nppiHistogramRangeGetBufferSize\_16u\_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.22 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.23 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16u\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.24 NppStatus nppiHistogramRangeGetBufferSize\_16u\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_16u\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.25 NppStatus nppiHistogramRangeGetBufferSize\_32f\_AC4R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_32f\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.26 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C1R (NppiSize *oSizeROI*, int *nLevels*, int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_32f\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.27 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)**

Scratch-buffer size for nppiHistogramRange\_32f\_C3R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.28 NppStatus nppiHistogramRangeGetBufferSize\_32f\_C4R (NppiSize oSizeROI, int nLevels[4], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_32f\_C4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.29 NppStatus nppiHistogramRangeGetBufferSize\_8u\_AC4R (NppiSize oSizeROI, int nLevels[3], int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_8u\_AC4R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

**7.115.2.30 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C1R (NppiSize oSizeROI, int nLevels, int \* hpBufferSize)**

Scratch-buffer size for nppiHistogramRange\_8u\_C1R.

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Number of levels in the histogram.

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#)..

### 7.115.2.31 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C3R (NppiSize *oSizeROI*, int *nLevels*[3], int \* *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange\_8u\_C3R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

### 7.115.2.32 NppStatus nppiHistogramRangeGetBufferSize\_8u\_C4R (NppiSize *oSizeROI*, int *nLevels*[4], int \* *hpBufferSize*)

Scratch-buffer size for nppiHistogramRange\_8u\_C4R.

#### Parameters:

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nLevels* Array containing number of levels per color channel.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#)..

## 7.116 Image Proximity

Primitives for computing the proximity measure between a source image and a template image.

### Modules

- [SqrDistanceFull\\_Norm](#)

*Primitives for computing the normalized Euclidean distance between two images with full mode.*

- [SqrDistanceSame\\_Norm](#)

*Primitives for computing the normalized Euclidean distance between two images with same mode.*

- [SqrDistanceValid\\_Norm](#)

*Primitives for computing the normalized Euclidean distance between two images with valid mode.*

- [CrossCorrFull\\_Norm](#)

*Primitives for computing the normalized cross correlation between two images with full mode.*

- [CrossCorrSame\\_Norm](#)

*Primitives for computing the normalized cross correlation between two images with same mode.*

- [CrossCorrValid\\_Norm](#)

*Primitives for computing the normalized cross correlation between two images with valid mode.*

- [CrossCorrValid](#)

*Primitives for computing the cross correlation between two images with valid mode.*

- [CrossCorrFull\\_NormLevel](#)

*Primitives for computing the normalized cross correlation coefficient between two images with full mode.*

- [CrossCorrSame\\_NormLevel](#)

*Primitives for computing the normalized cross correlation coefficient between two images with same mode.*

- [CrossCorrValid\\_NormLevel](#)

*Primitives for computing the normalized cross correlation coefficient between two images with valid mode.*

### 7.116.1 Detailed Description

Primitives for computing the proximity measure between a source image and a template image.

### 7.116.2 General Introduction

There are basically two approaches to compute the proximity measure for template matching, Euclidean distance and the cross correlation.

1. Euclidean distance computes the sum of the squared distance (SSD) between the corresponding pixels of the source image and the template image. The smaller the distance is, the more similar the source image and the template image is around the pixel. The anchor of the template image is used during the computations, which always lies in the geometric center of the image. Given a source image  $pSrc$  ( $W_s \times H_s$ ) and a template image  $pTpl$  ( $W_t \times H_t$ ), the Euclidean distance  $D_{st}(c, r)$  between two images at pixel in row  $r$  and column  $c$  is computed as ( $s$  stands for source image and  $t$  for template image for short):

$$D_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]^2$$

2. Cross correlation computes the sum of the product between the corresponding pixels of the source image and the template image. The cross correlation  $R_{st}(c, r)$  is calculated as:

$$R_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) \cdot pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2})]$$

The larger the cross correlation value is, the more similar the source image and the template image is around the pixel.

3. The cross correlation  $R_{st}(c, r)$  is affected by the brightness of the images which may vary due to the lighting and exposure conditions. Therefore, NPP computes the cross correlation coefficient to circumvent this dependence. This is typically done at every step by subtracting the mean from every pixel value, i.e.,

$$\tilde{R}_{st}(c, r) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t] \cdot [pSrc(j + c - \frac{H_t}{2}, i + r - \frac{W_t}{2}) - Mean_s]$$

NPP computes the normalized values of Euclidean distance, cross correlation and the cross correlation coefficient.

1. The normalized Euclidean distance  $\sigma_{st}(c, r)$  is defined as:

$$\sigma_{st}(c, r) = \frac{D_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

2. The normalized cross correlation  $\rho_{st}(c, r)$  is defined as:

$$\rho_{st}(c, r) = \frac{R_{st}(c, r)}{\sqrt{R_{ss}(c, r) \cdot R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The  $R_{ss}(c, r)$  and  $R_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$  denote the auto correlation of the source image and the template image individually. They are defined as:

$$R_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} pSrc(j, i)$$

$$R_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} pTpl(j, i)$$



3. Similarly, the normalized cross correlation coefficient  $\gamma_{st}(c, r)$  is calculated as:

$$\gamma_{st}(c, r) = \frac{\tilde{R}_{st}(c, r)}{\sqrt{\tilde{R}_{ss}(c, r) \cdot \tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})}}$$

The  $\tilde{R}_{ss}(c, r)$  and  $\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2})$  are defined as:

$$\tilde{R}_{ss}(c, r) = \sum_{j=c-\frac{H_t}{2}}^{c+\frac{H_t}{2}} \sum_{i=r-\frac{W_t}{2}}^{r+\frac{W_t}{2}} [pSrc(j, i) - Mean_s]$$

$$\tilde{R}_{tt}(\frac{H_t}{2}, \frac{W_t}{2}) = \sum_{j=0}^{H_t-1} \sum_{i=0}^{W_t-1} [pTpl(j, i) - Mean_t]$$

### 7.116.3 Categorizations

The Euclidean distance and the cross correlation are categorized into three types, full, same, and valid.

1. Full mode indicates that the anchor of the template image starts from the outside of the source image, assuming the out-of-boundary pixels are zero-padded. The size of the destination image is  $(W_s + W_t - 1) \times (H_s + H_t - 1)$ .
2. Same mode means that the anchor of the template image starts from the top left pixel of the source image. All the out-of-boundary pixels are also zero-padded. The size of the destination image is the same as the source one, i.e.,  $W_s \times H_s$ .
3. Valid mode indicates that there are no out-of-boundary readings from the source image. The anchor of the template image starts from the inside of the source image. The size of the destination image is  $(W_s - W_t + 1) \times (H_s - H_t + 1)$ .

## 7.117 SqrDistanceFull\_Norm

Primitives for computing the normalized Euclidean distance between two images with full mode.

### SqrDistanceFull\_Norm

The functions compute the  $\sigma_{st}(c, r)$  in [General Introduction](#) with full mode (see [Categorizations](#)).

- [NppStatus nppiSqrDistanceFull\\_Norm\\_8u\\_C1RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*One-channel 8-bit unsigned image SqrDistanceFull\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_8u\\_C3RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Three-channel 8-bit unsigned image SqrDistanceFull\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_8u\\_C4RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image SqrDistanceFull\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_8u\\_AC4RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image SqrDistanceFull\_Norm ignoring alpha channel, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 32-bit floating point image SqrDistanceFull\_Norm.*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Three-channel 32-bit floating point image SqrDistanceFull\_Norm.*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image SqrDistanceFull\_Norm.*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image SqrDistanceFull\_Norm ignoring alpha channel.*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_8u32f\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 8-bit unsigned image SqrDistanceFull\_Norm.*
- [NppStatus nppiSqrDistanceFull\\_Norm\\_8u32f\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)

*Three-channel 8-bit unsigned image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image SqrDistanceFull\_Norm ignoring alpha channel.*

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 8-bit signed image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 8-bit signed image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image SqrDistanceFull\_Norm ignoring alpha channel.*

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 16-bit unsigned image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 16-bit unsigned image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image SqrDistanceFull\_Norm.*

- `NppStatus nppiSqrDistanceFull_Norm_16u32f_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image SqrDistanceFull\_Norm ignoring alpha channel.*

### 7.117.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with full mode.

## 7.117.2 Function Documentation

**7.117.2.1** `NppStatus nppiSqrDistanceFull_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceFull\_Norm ignoring alpha channel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.2** `NppStatus nppiSqrDistanceFull_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image SqrDistanceFull\_Norm.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.3** `NppStatus nppiSqrDistanceFull_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.4** `NppStatus nppiSqrDistanceFull_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.5** `NppStatus nppiSqrDistanceFull_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceFull\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.117.2.6** `NppStatus nppiSqrDistanceFull_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.117.2.7** `NppStatus nppiSqrDistanceFull_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.8** `NppStatus nppiSqrDistanceFull_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.9** `NppStatus nppiSqrDistanceFull_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceFull\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.10** `NppStatus nppiSqrDistanceFull_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.11** `NppStatus nppiSqrDistanceFull_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.12** `NppStatus nppiSqrDistanceFull_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceFull\_Norm.



**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.117.2.13** `NppStatus nppiSqrDistanceFull_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceFull\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.117.2.14** `NppStatus nppiSqrDistanceFull_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.15** `NppStatus nppiSqrDistanceFull_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.16** `NppStatus nppiSqrDistanceFull_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceFull\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.17** `NppStatus nppiSqrDistanceFull_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceFull\_Norm ignoring alpha channel, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.18** `NppStatus nppiSqrDistanceFull_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceFull\_Norm, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.19** `NppStatus nppiSqrDistanceFull_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceFull\_Norm, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.117.2.20** `NppStatus nppiSqrDistanceFull_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceFull\_Norm, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.118 SqrDistanceSame\_Norm

Primitives for computing the normalized Euclidean distance between two images with same mode.

### SqrDistanceSame\_Norm

The functions compute the  $\sigma_{st}(c, r)$  in [General Introduction](#) with same mode (see [Categorizations](#)).

- `NppStatus nppiSqrDistanceSame_Norm_8u_C1RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)  
*One-channel 8-bit unsigned image SqrDistanceSame\_Norm, scaled by  $2^{\ell} - nScaleFactor$ .*
- `NppStatus nppiSqrDistanceSame_Norm_8u_C3RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)  
*Three-channel 8-bit unsigned image SqrDistanceSame\_Norm, scaled by  $2^{\ell} - nScaleFactor$ .*
- `NppStatus nppiSqrDistanceSame_Norm_8u_C4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image SqrDistanceSame\_Norm, scaled by  $2^{\ell} - nScaleFactor$ .*
- `NppStatus nppiSqrDistanceSame_Norm_8u_AC4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image SqrDistanceSame\_Norm ignoring alpha channel, scaled by  $2^{\ell} - nScaleFactor$ .*
- `NppStatus nppiSqrDistanceSame_Norm_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*One-channel 32-bit floating point image SqrDistanceSame\_Norm.*
- `NppStatus nppiSqrDistanceSame_Norm_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*Three-channel 32-bit floating point image SqrDistanceSame\_Norm.*
- `NppStatus nppiSqrDistanceSame_Norm_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image SqrDistanceSame\_Norm.*
- `NppStatus nppiSqrDistanceSame_Norm_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image SqrDistanceSame\_Norm ignoring alpha channel.*
- `NppStatus nppiSqrDistanceSame_Norm_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*One-channel 8-bit unsigned image SqrDistanceSame\_Norm.*

- [NppStatus nppiSqrDistanceSame\\_Norm\\_8u32f\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Three-channel 8-bit unsigned image SqrDistanceSame\_Norm.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_8u32f\\_C4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 8-bit unsigned image SqrDistanceSame\_Norm.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_8u32f\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 8-bit unsigned image SqrDistanceSame\_Norm ignoring alpha channel.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_8s32f\\_C1R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 8-bit signed image SqrDistanceSame\_Norm.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_8s32f\\_C3R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Three-channel 8-bit signed image SqrDistanceSame\_Norm.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_8s32f\\_C4R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 8-bit signed image SqrDistanceSame\_Norm.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_8s32f\\_AC4R](#) (const [Npp8s](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8s](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 8-bit signed image SqrDistanceSame\_Norm ignoring alpha channel.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_16u32f\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 16-bit unsigned image SqrDistanceSame\_Norm.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_16u32f\\_C3R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Three-channel 16-bit unsigned image SqrDistanceSame\_Norm.*
- [NppStatus nppiSqrDistanceSame\\_Norm\\_16u32f\\_C4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 16-bit unsigned image SqrDistanceSame\_Norm.*

- **NppStatus nppiSqrDistanceSame\_Norm\_16u32f\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image SqrDistanceSame\_Norm ignoring alpha channel.*

### 7.118.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with same mode.

### 7.118.2 Function Documentation

- 7.118.2.1 NppStatus nppiSqrDistanceSame\_Norm\_16u32f\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceSame\_Norm ignoring alpha channel.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**oSrcRoiSize** Region-of-Interest (ROI).  
**pTpl** Pointer to the template image.  
**nTplStep** Number of bytes between successive rows in the template image.  
**oTplRoiSize** Region-of-Interest (ROI).  
**pDst** Destination-Image Pointer.  
**nDstStep** Destination-Image Line Step.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.118.2.2 NppStatus nppiSqrDistanceSame\_Norm\_16u32f\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceSame\_Norm.

#### Parameters:

**pSrc** Source-Image Pointer.  
**nSrcStep** Source-Image Line Step.  
**oSrcRoiSize** Region-of-Interest (ROI).  
**pTpl** Pointer to the template image.  
**nTplStep** Number of bytes between successive rows in the template image.  
**oTplRoiSize** Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.3** `NppStatus nppiSqrDistanceSame_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.4** `NppStatus nppiSqrDistanceSame_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.118.2.5** `NppStatus nppiSqrDistanceSame_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceSame\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.118.2.6** `NppStatus nppiSqrDistanceSame_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.118.2.7** `NppStatus nppiSqrDistanceSame_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.8** `NppStatus nppiSqrDistanceSame_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.9** `NppStatus nppiSqrDistanceSame_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceSame\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.10** `NppStatus nppiSqrDistanceSame_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.11** `NppStatus nppiSqrDistanceSame_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.12** `NppStatus nppiSqrDistanceSame_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.118.2.13** `NppStatus nppiSqrDistanceSame_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceSame\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.118.2.14** `NppStatus nppiSqrDistanceSame_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.15** `NppStatus nppiSqrDistanceSame_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.16** `NppStatus nppiSqrDistanceSame_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* [Region-of-Interest \(ROI\)](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.118.2.17** `NppStatus nppiSqrDistanceSame_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceSame\_Norm ignoring alpha channel, scaled by  $2^{\ell - nScaleFactor}$ .

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).  
*pTpl* [Pointer to the template image](#).  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* [Region-of-Interest \(ROI\)](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.118.2.18** `NppStatus nppiSqrDistanceSame_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceSame\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).  
*pTpl* [Pointer to the template image](#).  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* [Region-of-Interest \(ROI\)](#).  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.19** `NppStatus nppiSqrDistanceSame_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceSame\_Norm, scaled by  $2^{\text{--}nScaleFactor}$ .

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.118.2.20** `NppStatus nppiSqrDistanceSame_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceSame\_Norm, scaled by  $2^{\text{--}nScaleFactor}$ .

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.119 SqrDistanceValid\_Norm

Primitives for computing the normalized Euclidean distance between two images with valid mode.

### SqrDistanceValid\_Norm

The functions compute the  $\sigma_{st}(c, r)$  in [General Introduction](#) with valid mode (see [Categorizations](#)).

- `NppStatus nppiSqrDistanceValid_Norm_8u_C1RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*One-channel 8-bit unsigned image SqrDistanceValid\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*

- `NppStatus nppiSqrDistanceValid_Norm_8u_C3RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*Three-channel 8-bit unsigned image SqrDistanceValid\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*

- `NppStatus nppiSqrDistanceValid_Norm_8u_C4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*Four-channel 8-bit unsigned image SqrDistanceValid\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*

- `NppStatus nppiSqrDistanceValid_Norm_8u_AC4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*Four-channel 8-bit unsigned image SqrDistanceValid\_Norm ignoring alpha channel, scaled by  $2^{\ell - nScaleFactor}$ .*

- `NppStatus nppiSqrDistanceValid_Norm_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 32-bit floating point image SqrDistanceValid\_Norm.*

- `NppStatus nppiSqrDistanceValid_Norm_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 32-bit floating point image SqrDistanceValid\_Norm.*

- `NppStatus nppiSqrDistanceValid_Norm_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 32-bit floating point image SqrDistanceValid\_Norm.*

- `NppStatus nppiSqrDistanceValid_Norm_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 32-bit floating point image SqrDistanceValid\_Norm ignoring alpha channel.*

- `NppStatus nppiSqrDistanceValid_Norm_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 8-bit unsigned image SqrDistanceValid\_Norm.*



- **NppStatus** **nppiSqrDistanceValid\_Norm\_8u32f\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Three-channel 8-bit unsigned image SqrDistanceValid\_Norm.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_8u32f\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image SqrDistanceValid\_Norm.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_8u32f\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image SqrDistanceValid\_Norm ignoring alpha channel.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_8s32f\_C1R** (const **Npp8s** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*One-channel 8-bit signed image SqrDistanceValid\_Norm.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_8s32f\_C3R** (const **Npp8s** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Three-channel 8-bit signed image SqrDistanceValid\_Norm.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_8s32f\_C4R** (const **Npp8s** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Four-channel 8-bit signed image SqrDistanceValid\_Norm.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_8s32f\_AC4R** (const **Npp8s** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp8s** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Four-channel 8-bit signed image SqrDistanceValid\_Norm ignoring alpha channel.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_16u32f\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*One-channel 16-bit unsigned image SqrDistanceValid\_Norm.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_16u32f\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Three-channel 16-bit unsigned image SqrDistanceValid\_Norm.*

- **NppStatus** **nppiSqrDistanceValid\_Norm\_16u32f\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image SqrDistanceValid\_Norm.*

- [NppStatus nppiSqrDistanceValid\\_Norm\\_16u32f\\_AC4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image SqrDistanceValid\_Norm ignoring alpha channel.*

### 7.119.1 Detailed Description

Primitives for computing the normalized Euclidean distance between two images with valid mode.

### 7.119.2 Function Documentation

- 7.119.2.1** [NppStatus nppiSqrDistanceValid\\_Norm\\_16u32f\\_AC4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)

Four-channel 16-bit unsigned image SqrDistanceValid\_Norm ignoring alpha channel.

#### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSrcRoiSize](#) [Region-of-Interest \(ROI\)](#).

[pTpl](#) [Pointer to the template image](#).

[nTplStep](#) [Number of bytes between successive rows in the template image](#).

[oTplRoiSize](#) [Region-of-Interest \(ROI\)](#).

[pDst](#) [Destination-Image Pointer](#).

[nDstStep](#) [Destination-Image Line Step](#).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

- 7.119.2.2** [NppStatus nppiSqrDistanceValid\\_Norm\\_16u32f\\_C1R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp16u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)

One-channel 16-bit unsigned image SqrDistanceValid\_Norm.

#### Parameters:

[pSrc](#) [Source-Image Pointer](#).

[nSrcStep](#) [Source-Image Line Step](#).

[oSrcRoiSize](#) [Region-of-Interest \(ROI\)](#).

[pTpl](#) [Pointer to the template image](#).

[nTplStep](#) [Number of bytes between successive rows in the template image](#).

[oTplRoiSize](#) [Region-of-Interest \(ROI\)](#).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.3** `NppStatus nppiSqrDistanceValid_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.4** `NppStatus nppiSqrDistanceValid_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.5** `NppStatus nppiSqrDistanceValid_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceValid\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.119.2.6** `NppStatus nppiSqrDistanceValid_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.119.2.7** `NppStatus nppiSqrDistanceValid_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.8** `NppStatus nppiSqrDistanceValid_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.9** `NppStatus nppiSqrDistanceValid_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceValid\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.119.2.10** `NppStatus nppiSqrDistanceValid_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.119.2.11** `NppStatus nppiSqrDistanceValid_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.119.2.12** `NppStatus nppiSqrDistanceValid_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.119.2.13** `NppStatus nppiSqrDistanceValid_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceValid\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.119.2.14** `NppStatus nppiSqrDistanceValid_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.15** `NppStatus nppiSqrDistanceValid_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.16** `NppStatus nppiSqrDistanceValid_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image SqrDistanceValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.



*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.17** `NppStatus nppiSqrDistanceValid_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceValid\_Norm ignoring alpha channel, scaled by  $2^{\ell - nScaleFactor}$ .

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.18** `NppStatus nppiSqrDistanceValid_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image SqrDistanceValid\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.19** `NppStatus nppiSqrDistanceValid_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image SqrDistanceValid\_Norm, scaled by  $2^{\text{--}nScaleFactor}$ .

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.119.2.20** `NppStatus nppiSqrDistanceValid_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image SqrDistanceValid\_Norm, scaled by  $2^{\text{--}nScaleFactor}$ .

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.120 CrossCorrFull\_Norm

Primitives for computing the normalized cross correlation between two images with full mode.

### CrossCorrFull\_Norm

The functions compute the  $\rho_{st}(c, r)$  in [General Introduction](#) with full mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrFull_Norm_8u_C1RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*One-channel 8-bit unsigned image CrossCorrFull\_Norm, scaled by  $2^{\ell} - nScaleFactor$ .*

- `NppStatus nppiCrossCorrFull_Norm_8u_C3RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*Three-channel 8-bit unsigned image CrossCorrFull\_Norm, scaled by  $2^{\ell} - nScaleFactor$ .*

- `NppStatus nppiCrossCorrFull_Norm_8u_C4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*Four-channel 8-bit unsigned image CrossCorrFull\_Norm, scaled by  $2^{\ell} - nScaleFactor$ .*

- `NppStatus nppiCrossCorrFull_Norm_8u_AC4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor)

*Four-channel 8-bit unsigned image CrossCorrFull\_Norm ignoring alpha channel, scaled by  $2^{\ell} - nScaleFactor$ .*

- `NppStatus nppiCrossCorrFull_Norm_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 32-bit floating point image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 32-bit floating point image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 32-bit floating point image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 32-bit floating point image CrossCorrFull\_Norm ignoring alpha channel.*

- `NppStatus nppiCrossCorrFull_Norm_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 8-bit unsigned image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_8u32f_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 8-bit unsigned image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image CrossCorrFull\_Norm ignoring alpha channel.*

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 8-bit signed image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 8-bit signed image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image CrossCorrFull\_Norm ignoring alpha channel.*

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 16-bit unsigned image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 16-bit unsigned image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_16u32f_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image CrossCorrFull\_Norm.*

- `NppStatus nppiCrossCorrFull_Norm_16u32f_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image CrossCorrFull\_Norm ignoring alpha channel.*

### 7.120.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with full mode.

## 7.120.2 Function Documentation

**7.120.2.1** `NppStatus nppiCrossCorrFull_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrFull\_Norm ignoring alpha channel.

### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.2** `NppStatus nppiCrossCorrFull_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrFull\_Norm.

### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.3** `NppStatus nppiCrossCorrFull_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.4** `NppStatus nppiCrossCorrFull_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.5** `NppStatus nppiCrossCorrFull_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrFull\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.120.2.6** `NppStatus nppiCrossCorrFull_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.120.2.7** `NppStatus nppiCrossCorrFull_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.8** `NppStatus nppiCrossCorrFull_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.9** `NppStatus nppiCrossCorrFull_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrFull\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.120.2.10** `NppStatus nppiCrossCorrFull_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.11** `NppStatus nppiCrossCorrFull_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.12** `NppStatus nppiCrossCorrFull_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.120.2.13** `NppStatus nppiCrossCorrFull_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrFull\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.120.2.14** `NppStatus nppiCrossCorrFull_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.15** `NppStatus nppiCrossCorrFull_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.16** `NppStatus nppiCrossCorrFull_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrFull\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.17** `NppStatus nppiCrossCorrFull_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrFull\_Norm ignoring alpha channel, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).  
*pTpl* [Pointer to the template image](#).  
*nTplStep* [Number of bytes between successive rows in the template image](#).  
*oTplRoiSize* [Region-of-Interest \(ROI\)](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.18** `NppStatus nppiCrossCorrFull_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrFull\_Norm, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).  
*pTpl* [Pointer to the template image](#).  
*nTplStep* [Number of bytes between successive rows in the template image](#).  
*oTplRoiSize* [Region-of-Interest \(ROI\)](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.120.2.19** `NppStatus nppiCrossCorrFull_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrFull\_Norm, scaled by  $2^{(-nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.120.2.20** `NppStatus nppiCrossCorrFull_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrFull\_Norm, scaled by  $2^{(-nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.121 CrossCorrSame\_Norm

Primitives for computing the normalized cross correlation between two images with same mode.

### CrossCorrSame\_Norm

The functions compute the  $\rho_{st}(c, r)$  in [General Introduction](#) with same mode (see [Categorizations](#)).

- [NppStatus nppiCrossCorrSame\\_Norm\\_8u\\_C1RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*One-channel 8-bit unsigned image CrossCorrSame\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrSame\\_Norm\\_8u\\_C3RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Three-channel 8-bit unsigned image CrossCorrSame\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrSame\\_Norm\\_8u\\_C4RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image CrossCorrSame\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrSame\\_Norm\\_8u\\_AC4RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image CrossCorrSame\_Norm ignoring alpha channel, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrSame\\_Norm\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 32-bit floating point image CrossCorrSame\_Norm.*
- [NppStatus nppiCrossCorrSame\\_Norm\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Three-channel 32-bit floating point image CrossCorrSame\_Norm.*
- [NppStatus nppiCrossCorrSame\\_Norm\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image CrossCorrSame\_Norm.*
- [NppStatus nppiCrossCorrSame\\_Norm\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image CrossCorrSame\_Norm ignoring alpha channel.*
- [NppStatus nppiCrossCorrSame\\_Norm\\_8u32f\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 8-bit unsigned image CrossCorrSame\_Norm.*
- [NppStatus nppiCrossCorrSame\\_Norm\\_8u32f\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)

*Three-channel 8-bit unsigned image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image CrossCorrSame\_Norm ignoring alpha channel.*

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 8-bit signed image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 8-bit signed image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image CrossCorrSame\_Norm ignoring alpha channel.*

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 16-bit unsigned image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 16-bit unsigned image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_16u32f_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image CrossCorrSame\_Norm.*

- `NppStatus nppiCrossCorrSame_Norm_16u32f_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image CrossCorrSame\_Norm ignoring alpha channel.*

### 7.121.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with same mode.

## 7.121.2 Function Documentation

**7.121.2.1** `NppStatus nppiCrossCorrSame_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrSame\_Norm ignoring alpha channel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.2** `NppStatus nppiCrossCorrSame_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrSame\_Norm.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.121.2.3** `NppStatus nppiCrossCorrSame_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.4** `NppStatus nppiCrossCorrSame_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.5** `NppStatus nppiCrossCorrSame_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrSame\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.6** `NppStatus nppiCrossCorrSame_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.7** `NppStatus nppiCrossCorrSame_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.8** `NppStatus nppiCrossCorrSame_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.9** `NppStatus nppiCrossCorrSame_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrSame\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.10** `NppStatus nppiCrossCorrSame_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.11** `NppStatus nppiCrossCorrSame_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.12** `NppStatus nppiCrossCorrSame_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.13** `NppStatus nppiCrossCorrSame_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrSame\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.121.2.14** `NppStatus nppiCrossCorrSame_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.15** `NppStatus nppiCrossCorrSame_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.16** `NppStatus nppiCrossCorrSame_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrSame\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.17** `NppStatus nppiCrossCorrSame_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrSame\_Norm ignoring alpha channel, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.18** `NppStatus nppiCrossCorrSame_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrSame\_Norm, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.19** `NppStatus nppiCrossCorrSame_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrSame\_Norm, scaled by  $2^{\lfloor -nScaleFactor \rfloor}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.121.2.20** `NppStatus nppiCrossCorrSame_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrSame\_Norm, scaled by  $2^{\lfloor -nScaleFactor \rfloor}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



## 7.122 CrossCorrValid\_Norm

Primitives for computing the normalized cross correlation between two images with valid mode.

### CrossCorrValid\_Norm

The functions compute the  $\rho_{st}(c, r)$  in [General Introduction](#) with valid mode (see [Categorizations](#)).

- [NppStatus nppiCrossCorrValid\\_Norm\\_8u\\_C1RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*One-channel 8-bit unsigned image CrossCorrValid\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrValid\\_Norm\\_8u\\_C3RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Three-channel 8-bit unsigned image CrossCorrValid\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrValid\\_Norm\\_8u\\_C4RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image CrossCorrValid\_Norm, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrValid\\_Norm\\_8u\\_AC4RSfs](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp8u](#) \*pDst, int nDstStep, int nScaleFactor)  
*Four-channel 8-bit unsigned image CrossCorrValid\_Norm ignoring alpha channel, scaled by  $2^{\ell - nScaleFactor}$ .*
- [NppStatus nppiCrossCorrValid\\_Norm\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 32-bit floating point image CrossCorrValid\_Norm.*
- [NppStatus nppiCrossCorrValid\\_Norm\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Three-channel 32-bit floating point image CrossCorrValid\_Norm.*
- [NppStatus nppiCrossCorrValid\\_Norm\\_32f\\_C4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image CrossCorrValid\_Norm.*
- [NppStatus nppiCrossCorrValid\\_Norm\\_32f\\_AC4R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp32f](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*Four-channel 32-bit floating point image CrossCorrValid\_Norm ignoring alpha channel.*
- [NppStatus nppiCrossCorrValid\\_Norm\\_8u32f\\_C1R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)  
*One-channel 8-bit unsigned image CrossCorrValid\_Norm.*
- [NppStatus nppiCrossCorrValid\\_Norm\\_8u32f\\_C3R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [NppiSize](#) oSrcRoiSize, const [Npp8u](#) \*pTpl, int nTplStep, [NppiSize](#) oTplRoiSize, [Npp32f](#) \*pDst, int nDstStep)

*Three-channel 8-bit unsigned image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit unsigned image CrossCorrValid\_Norm ignoring alpha channel.*

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 8-bit signed image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 8-bit signed image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 8-bit signed image CrossCorrValid\_Norm ignoring alpha channel.*

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*One-channel 16-bit unsigned image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Three-channel 16-bit unsigned image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_16u32f_C4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image CrossCorrValid\_Norm.*

- `NppStatus nppiCrossCorrValid_Norm_16u32f_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

*Four-channel 16-bit unsigned image CrossCorrValid\_Norm ignoring alpha channel.*

### 7.122.1 Detailed Description

Primitives for computing the normalized cross correlation between two images with valid mode.

## 7.122.2 Function Documentation

**7.122.2.1** `NppStatus nppiCrossCorrValid_Norm_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrValid\_Norm ignoring alpha channel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.2** `NppStatus nppiCrossCorrValid_Norm_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 16-bit unsigned image CrossCorrValid\_Norm.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.3** `NppStatus nppiCrossCorrValid_Norm_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 16-bit unsigned image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.4** `NppStatus nppiCrossCorrValid_Norm_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 16-bit unsigned image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.5** `NppStatus nppiCrossCorrValid_Norm_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrValid\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.6** `NppStatus nppiCrossCorrValid_Norm_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.7** `NppStatus nppiCrossCorrValid_Norm_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 32-bit floating point image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.8** `NppStatus nppiCrossCorrValid_Norm_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 32-bit floating point image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.9** `NppStatus nppiCrossCorrValid_Norm_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrValid\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.10** `NppStatus nppiCrossCorrValid_Norm_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.11** `NppStatus nppiCrossCorrValid_Norm_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit signed image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.12** `NppStatus nppiCrossCorrValid_Norm_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit signed image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.13** `NppStatus nppiCrossCorrValid_Norm_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrValid\_Norm ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.14** `NppStatus nppiCrossCorrValid_Norm_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.



*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.15** `NppStatus nppiCrossCorrValid_Norm_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Three-channel 8-bit unsigned image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.16** `NppStatus nppiCrossCorrValid_Norm_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

Four-channel 8-bit unsigned image CrossCorrValid\_Norm.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.17** `NppStatus nppiCrossCorrValid_Norm_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrValid\_Norm ignoring alpha channel, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.18** `NppStatus nppiCrossCorrValid_Norm_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

One-channel 8-bit unsigned image CrossCorrValid\_Norm, scaled by  $2^{( - nScaleFactor)}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.122.2.19** `NppStatus nppiCrossCorrValid_Norm_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Three-channel 8-bit unsigned image CrossCorrValid\_Norm, scaled by  $2^{\lceil -nScaleFactor \rceil}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.122.2.20** `NppStatus nppiCrossCorrValid_Norm_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor)`

Four-channel 8-bit unsigned image CrossCorrValid\_Norm, scaled by  $2^{\lceil -nScaleFactor \rceil}$ .

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*oSrcRoiSize* Region-of-Interest (ROI).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* Region-of-Interest (ROI).  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

## 7.123 CrossCorrValid

Primitives for computing the cross correlation between two images with valid mode.

### CrossCorrValid

The functions compute the  $R_{st}(c, r)$  in [General Introduction](#) with valid mode (see [Categorizations](#)).

- `NppStatus nppiCrossCorrValid_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*One-channel 32-bit floating point images CrossCorrValid.*
- `NppStatus nppiCrossCorrValid_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*One-channel 8-bit unsigned images CrossCorrValid.*
- `NppStatus nppiCrossCorrValid_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*One-channel 8-bit signed images CrossCorrValid.*
- `NppStatus nppiCrossCorrValid_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)  
*One-channel 16-bit unsigned images CrossCorrValid.*

### 7.123.1 Detailed Description

Primitives for computing the cross correlation between two images with valid mode.

### 7.123.2 Function Documentation

**7.123.2.1** `NppStatus nppiCrossCorrValid_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep)

One-channel 16-bit unsigned images CrossCorrValid.

#### Parameters:

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcRoiSize* [Region-of-Interest \(ROI\)](#).
- pTpl* [Pointer to the template image](#).
- nTplStep* [Number of bytes between successive rows in the template image](#).
- oTplRoiSize* [Region-of-Interest \(ROI\)](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.123.2.2** `NppStatus nppiCrossCorrValid_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 32-bit floating point images CrossCorrValid.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.123.2.3** `NppStatus nppiCrossCorrValid_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit signed images CrossCorrValid.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.123.2.4** `NppStatus nppiCrossCorrValid_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep)`

One-channel 8-bit unsigned images CrossCorrValid.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.124 CrossCorrFull\_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

### CrossCorrFull\_NormLevel

The functions compute the  $\gamma_{st}(c, r)$  in [General Introduction](#) with full mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrFull_NormLevel_8u_C1RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit unsigned image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8u_C3RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit unsigned image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8u_C4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8u_AC4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrFull_NormLevel_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 32-bit floating point image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 32-bit floating point image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image CrossCorrFull\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit unsigned image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit unsigned image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit signed image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit signed image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit signed image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit signed image CrossCorrFull\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 16-bit unsigned image CrossCorrFull\_NormLevel.*

- `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 16-bit unsigned image CrossCorrFull\_NormLevel.*



- **NppStatus** **nppiCrossCorrFull\_NormLevel\_16u32f\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit unsigned image CrossCorrFull\_NormLevel.*

- **NppStatus** **nppiCrossCorrFull\_NormLevel\_16u32f\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit unsigned image CrossCorrFull\_NormLevel ignoring alpha channel.*

## FullNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrFull\_NormLevel primitives.

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_8u\_C1RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_8u\_C1RSfs.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_8u\_C3RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_8u\_C3RSfs.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_8u\_C4RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_8u\_C4RSfs.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_8u\_AC4RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_8u\_AC4RSfs.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_32f\_C1R.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_32f\_C3R.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_32f\_C4R.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrFull\_NormLevel\_32f\_AC4R.*

- **NppStatus** **nppiFullNormLevelGetBufferHostSize\_8u32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C1R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8u32f_C3R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C3R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8u32f_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_C4R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8u32f_AC4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8u32f_AC4R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_C1R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C1R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_C3R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C3R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_C4R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_8s32f_AC4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_8s32f_AC4R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_C1R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_C1R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_C3R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_C3R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_C4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_C4R`.*

- **NppStatus** `nppiFullNormLevelGetBufferHostSize_16u32f_AC4R` (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)

*Buffer size (in bytes) for `nppiCrossCorrFull_NormLevel_16u32f_AC4R`.*

### 7.124.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with full mode.

## 7.124.2 Function Documentation

**7.124.2.1** `NppStatus nppiCrossCorrFull_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrFull\_NormLevel ignoring alpha channel.

### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_16u32f\\_AC4R](#) to compute the required size (in bytes).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.2** `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrFull\_NormLevel.

### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_16u32f\\_C1R](#) to compute the required size (in bytes).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.3** `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_16u32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.4** `NppStatus nppiCrossCorrFull_NormLevel_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_16u32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.5** `NppStatus nppiCrossCorrFull_NormLevel_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrFull\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.6** `NppStatus nppiCrossCorrFull_NormLevel_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.7** `NppStatus nppiCrossCorrFull_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.8** `NppStatus nppiCrossCorrFull_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.9** `NppStatus nppiCrossCorrFull_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrFull\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8s32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.10** `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8s32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.11** `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8s32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.12** `NppStatus nppiCrossCorrFull_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8s32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.124.2.13** `NppStatus nppiCrossCorrFull_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.14** `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.15** `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.16** `NppStatus nppiCrossCorrFull_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.17** `NppStatus nppiCrossCorrFull_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u\\_AC4RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.18** `NppStatus nppiCrossCorrFull_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u\\_C1RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.19** `NppStatus nppiCrossCorrFull_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u\\_C3RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.20** `NppStatus nppiCrossCorrFull_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrFull\_NormLevel.

**Parameters:**

- pSrc* Source-Image Pointer.
- nSrcStep* Source-Image Line Step.
- oSrcRoiSize* Region-of-Interest (ROI).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* Region-of-Interest (ROI).
- pDst* Destination-Image Pointer.
- nDstStep* Destination-Image Line Step.
- nScaleFactor* Integer Result Scaling.
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiFullNormLevelGetBufferHostSize\\_8u\\_C4RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.124.2.21 NppStatus nppiFullNormLevelGetBufferHostSize\_16u32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_16u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.22 NppStatus nppiFullNormLevelGetBufferHostSize\_16u32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_16u32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.23 NppStatus nppiFullNormLevelGetBufferHostSize\_16u32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_16u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.24 NppStatus nppiFullNormLevelGetBufferHostSize\_16u32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_16u32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.25 NppStatus nppiFullNormLevelGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.26 NppStatus nppiFullNormLevelGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.27 NppStatus nppiFullNormLevelGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.28 NppStatus nppiFullNormLevelGetBufferHostSize\_32f\_C4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.29 NppStatus nppiFullNormLevelGetBufferHostSize\_8s32f\_AC4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8s32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.30 NppStatus nppiFullNormLevelGetBufferHostSize\_8s32f\_C1R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8s32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.31 NppStatus nppiFullNormLevelGetBufferHostSize\_8s32f\_C3R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8s32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.32 NppStatus nppiFullNormLevelGetBufferHostSize\_8s32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8s32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.33 NppStatus nppiFullNormLevelGetBufferHostSize\_8u32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.34 NppStatus nppiFullNormLevelGetBufferHostSize\_8u32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.124.2.35 NppStatus nppiFullNormLevelGetBufferHostSize\_8u32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.36 NppStatus nppiFullNormLevelGetBufferHostSize\_8u32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.37 NppStatus nppiFullNormLevelGetBufferHostSize\_8u\_AC4RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u\\_AC4RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.38 NppStatus nppiFullNormLevelGetBufferHostSize\_8u\_C1RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u\\_C1RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.39 NppStatus nppiFullNormLevelGetBufferHostSize\_8u\_C3RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u\\_C3RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.124.2.40 NppStatus nppiFullNormLevelGetBufferHostSize\_8u\_C4RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrFull\\_NormLevel\\_8u\\_C4RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.125 CrossCorrSame\_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

### CrossCorrSame\_NormLevel

The functions compute the  $\gamma_{st}(c, r)$  in [General Introduction](#) with same mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrSame_NormLevel_8u_C1RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit unsigned image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8u_C3RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit unsigned image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8u_C4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8u_AC4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrSame_NormLevel_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 32-bit floating point image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 32-bit floating point image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image CrossCorrSame\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit unsigned image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit unsigned image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit signed image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit signed image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit signed image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit signed image CrossCorrSame\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 16-bit unsigned image CrossCorrSame\_NormLevel.*

- `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 16-bit unsigned image CrossCorrSame\_NormLevel.*

- **NppStatus** **nppiCrossCorrSame\_NormLevel\_16u32f\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit unsigned image CrossCorrSame\_NormLevel.*

- **NppStatus** **nppiCrossCorrSame\_NormLevel\_16u32f\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit unsigned image CrossCorrSame\_NormLevel ignoring alpha channel.*

## SameNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrSame\_-NormLevel primitives.

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_8u\_C1RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_8u\_C1RSfs.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_8u\_C3RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_8u\_C3RSfs.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_8u\_C4RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_8u\_C4RSfs.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_8u\_AC4RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_8u\_AC4RSfs.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_32f\_C1R.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_32f\_C3R.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_32f\_C4R.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrSame\_NormLevel\_32f\_AC4R.*

- **NppStatus** **nppiSameNormLevelGetBufferHostSize\_8u32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C1R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C3R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_C4R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8u32f_AC4R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C1R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C3R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_C4R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_8s32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_8s32f_AC4R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_C1R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_C3R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_C4R`.*

- `NppStatus` `nppiSameNormLevelGetBufferHostSize_16u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrSame_NormLevel_16u32f_AC4R`.*

### 7.125.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with same mode.

## 7.125.2 Function Documentation

**7.125.2.1** `NppStatus nppiCrossCorrSame_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrSame\_NormLevel ignoring alpha channel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize\\_16u32f\\_AC4R](#) to compute the required size (in bytes).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.2** `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrSame\_NormLevel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiSameNormLevelGetBufferHostSize\\_16u32f\\_C1R](#) to compute the required size (in bytes).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.3** `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_16u32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.4** `NppStatus nppiCrossCorrSame_NormLevel_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_16u32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.125.2.5** `NppStatus nppiCrossCorrSame_NormLevel_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrSame\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.6** `NppStatus nppiCrossCorrSame_NormLevel_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.7** `NppStatus nppiCrossCorrSame_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.8** `NppStatus nppiCrossCorrSame_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.9** `NppStatus nppiCrossCorrSame_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrSame\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8s32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.10** `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8s32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.11** `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8s32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.12** `NppStatus nppiCrossCorrSame_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8s32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.13** `NppStatus nppiCrossCorrSame_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8u32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.14** `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8u32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.15** `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8u32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.16** `NppStatus nppiCrossCorrSame_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8u32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.17** `NppStatus nppiCrossCorrSame_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8u\\_AC4RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.18** `NppStatus nppiCrossCorrSame_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* [Pointer to the template image](#).

*nTplStep* [Number of bytes between successive rows in the template image](#).

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiSameNormLevelGetBufferHostSize\\_8u\\_C1RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.19** `NppStatus nppiCrossCorrSame_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* [Region-of-Interest \(ROI\)](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*nScaleFactor* [Integer Result Scaling](#).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiSameNormLevelGetBufferHostSize\\_8u\\_C3RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.125.2.20** `NppStatus nppiCrossCorrSame_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrSame\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).  
*pTpl* Pointer to the template image.  
*nTplStep* Number of bytes between successive rows in the template image.  
*oTplRoiSize* [Region-of-Interest \(ROI\)](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*nScaleFactor* [Integer Result Scaling](#).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiSameNormLevelGetBufferHostSize\\_8u\\_C4RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.125.2.21 NppStatus nppiSameNormLevelGetBufferHostSize\_16u32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_16u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.22 NppStatus nppiSameNormLevelGetBufferHostSize\_16u32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_16u32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.23 NppStatus nppiSameNormLevelGetBufferHostSize\_16u32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_16u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.24 NppStatus nppiSameNormLevelGetBufferHostSize\_16u32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_16u32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.25 NppStatus nppiSameNormLevelGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_32f\\_AC4R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.26 NppStatus nppiSameNormLevelGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_32f\\_C1R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.27 NppStatus nppiSameNormLevelGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_32f\\_C3R](#).

**Parameters:**

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.28 NppStatus nppiSameNormLevelGetBufferHostSize\_32f\_C4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.29 NppStatus nppiSameNormLevelGetBufferHostSize\_8s32f\_AC4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8s32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.30 NppStatus nppiSameNormLevelGetBufferHostSize\_8s32f\_C1R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8s32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.31 NppStatus nppiSameNormLevelGetBufferHostSize\_8s32f\_C3R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8s32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.32 NppStatus nppiSameNormLevelGetBufferHostSize\_8s32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8s32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.33 NppStatus nppiSameNormLevelGetBufferHostSize\_8u32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.34 NppStatus nppiSameNormLevelGetBufferHostSize\_8u32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.35 NppStatus nppiSameNormLevelGetBufferHostSize\_8u32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.36 NppStatus nppiSameNormLevelGetBufferHostSize\_8u32f\_C4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.37 NppStatus nppiSameNormLevelGetBufferHostSize\_8u\_AC4RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u\\_AC4RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.38 NppStatus nppiSameNormLevelGetBufferHostSize\_8u\_C1RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u\\_C1RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.39 NppStatus nppiSameNormLevelGetBufferHostSize\_8u\_C3RSfs (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u\\_C3RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.125.2.40 NppStatus nppiSameNormLevelGetBufferHostSize\_8u\_C4RSfs (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrSame\\_NormLevel\\_8u\\_C4RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.126 CrossCorrValid\_NormLevel

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

### CrossCorrValid\_NormLevel

The functions compute the  $\gamma_{st}(c, r)$  in [General Introduction](#) with valid mode (see [Categorizations](#)).

The functions require additional scratch buffer for computations.

- `NppStatus nppiCrossCorrValid_NormLevel_8u_C1RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit unsigned image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8u_C3RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit unsigned image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8u_C4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8u_AC4RSfs` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp8u` \*pDst, int nDstStep, int nScaleFactor, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrValid_NormLevel_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 32-bit floating point image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 32-bit floating point image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_32f_C4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp32f` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image CrossCorrValid\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit unsigned image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit unsigned image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8u32f_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C1R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 8-bit signed image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C3R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit signed image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit signed image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_8s32f_AC4R` (const `Npp8s` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp8s` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Four-channel 8-bit signed image CrossCorrValid\_NormLevel ignoring alpha channel.*

- `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*One-channel 16-bit unsigned image CrossCorrValid\_NormLevel.*

- `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `NppiSize` oSrcRoiSize, const `Npp16u` \*pTpl, int nTplStep, `NppiSize` oTplRoiSize, `Npp32f` \*pDst, int nDstStep, `Npp8u` \*pDeviceBuffer)

*Three-channel 16-bit unsigned image CrossCorrValid\_NormLevel.*



- **NppStatus** **nppiCrossCorrValid\_NormLevel\_16u32f\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit unsigned image CrossCorrValid\_NormLevel.*

- **NppStatus** **nppiCrossCorrValid\_NormLevel\_16u32f\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, **NppiSize** oSrcRoiSize, const **Npp16u** \*pTpl, int nTplStep, **NppiSize** oTplRoiSize, **Npp32f** \*pDst, int nDstStep, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit unsigned image CrossCorrValid\_NormLevel ignoring alpha channel.*

## ValidNormLevelGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the CrossCorrValid\_NormLevel primitives.

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_8u\_C1RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_8u\_C1RSfs.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_8u\_C3RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_8u\_C3RSfs.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_8u\_C4RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_8u\_C4RSfs.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_8u\_AC4RSfs** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_8u\_AC4RSfs.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_32f\_C1R.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_32f\_C3R.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_32f\_C4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_32f\_C4R.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for nppiCrossCorrValid\_NormLevel\_32f\_AC4R.*

- **NppStatus** **nppiValidNormLevelGetBufferHostSize\_8u32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_C1R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_C3R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_C4R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8u32f_AC4R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_C1R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_C3R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_C4R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_8s32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_8s32f_AC4R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_C1R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_C1R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_C3R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_C3R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_C4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_C4R`.*

- `NppStatus` `nppiValidNormLevelGetBufferHostSize_16u32f_AC4R` (`NppiSize` `oSizeROI`, `int` `*hpBufferSize`)

*Buffer size (in bytes) for `nppiCrossCorrValid_NormLevel_16u32f_AC4R`.*

### 7.126.1 Detailed Description

Primitives for computing the normalized cross correlation coefficient between two images with valid mode.

## 7.126.2 Function Documentation

**7.126.2.1** `NppStatus nppiCrossCorrValid_NormLevel_16u32f_AC4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrValid\_NormLevel ignoring alpha channel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_16u32f\\_AC4R](#) to compute the required size (in bytes).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.2** `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C1R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image CrossCorrValid\_NormLevel.

### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_16u32f\\_C1R](#) to compute the required size (in bytes).

### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.3** `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C3R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_16u32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.4** `NppStatus nppiCrossCorrValid_NormLevel_16u32f_C4R (const Npp16u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp16u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*oSrcRoiSize* Region-of-Interest (ROI).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* Region-of-Interest (ROI).

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_16u32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.5** `NppStatus nppiCrossCorrValid_NormLevel_32f_AC4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrValid\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.6** `NppStatus nppiCrossCorrValid_NormLevel_32f_C1R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.7** `NppStatus nppiCrossCorrValid_NormLevel_32f_C3R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.8** `NppStatus nppiCrossCorrValid_NormLevel_32f_C4R (const Npp32f * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp32f * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.9** `NppStatus nppiCrossCorrValid_NormLevel_8s32f_AC4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrValid\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8s32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.10** `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C1R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8s32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.11** `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C3R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8s32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.12** `NppStatus nppiCrossCorrValid_NormLevel_8s32f_C4R (const Npp8s * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8s * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8s32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.126.2.13** `NppStatus nppiCrossCorrValid_NormLevel_8u32f_AC4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.14** `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C1R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.15** `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C3R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.16** `NppStatus nppiCrossCorrValid_NormLevel_8u32f_C4R (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp32f * pDst, int nDstStep, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.17** `NppStatus nppiCrossCorrValid_NormLevel_8u_AC4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel ignoring alpha channel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u\\_AC4RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.18** `NppStatus nppiCrossCorrValid_NormLevel_8u_C1RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*oSrcRoiSize* [Region-of-Interest \(ROI\)](#).

*pTpl* Pointer to the template image.

*nTplStep* Number of bytes between successive rows in the template image.

*oTplRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u\\_C1RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.19** `NppStatus nppiCrossCorrValid_NormLevel_8u_C3RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcRoiSize* [Region-of-Interest \(ROI\)](#).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* [Region-of-Interest \(ROI\)](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- nScaleFactor* [Integer Result Scaling](#).
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u\\_C3RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.20** `NppStatus nppiCrossCorrValid_NormLevel_8u_C4RSfs (const Npp8u * pSrc, int nSrcStep, NppiSize oSrcRoiSize, const Npp8u * pTpl, int nTplStep, NppiSize oTplRoiSize, Npp8u * pDst, int nDstStep, int nScaleFactor, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image CrossCorrValid\_NormLevel.

**Parameters:**

- pSrc* [Source-Image Pointer](#).
- nSrcStep* [Source-Image Line Step](#).
- oSrcRoiSize* [Region-of-Interest \(ROI\)](#).
- pTpl* Pointer to the template image.
- nTplStep* Number of bytes between successive rows in the template image.
- oTplRoiSize* [Region-of-Interest \(ROI\)](#).
- pDst* [Destination-Image Pointer](#).
- nDstStep* [Destination-Image Line Step](#).
- nScaleFactor* [Integer Result Scaling](#).
- pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiValidNormLevelGetBufferHostSize\\_8u\\_C4RSfs](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.126.2.21 NppStatus nppiValidNormLevelGetBufferHostSize\_16u32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_16u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.22 NppStatus nppiValidNormLevelGetBufferHostSize\_16u32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_16u32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.23 NppStatus nppiValidNormLevelGetBufferHostSize\_16u32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_16u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.24 NppStatus nppiValidNormLevelGetBufferHostSize\_16u32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_16u32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.25 NppStatus nppiValidNormLevelGetBufferHostSize\_32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.26 NppStatus nppiValidNormLevelGetBufferHostSize\_32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.27 NppStatus nppiValidNormLevelGetBufferHostSize\_32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.28 NppStatus nppiValidNormLevelGetBufferHostSize\_32f\_C4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.29 NppStatus nppiValidNormLevelGetBufferHostSize\_8s32f\_AC4R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8s32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.30 NppStatus nppiValidNormLevelGetBufferHostSize\_8s32f\_C1R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8s32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.31 NppStatus nppiValidNormLevelGetBufferHostSize\_8s32f\_C3R** (NppiSize *oSizeROI*, int \* *hpBufferSize*)

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8s32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.32 NppStatus nppiValidNormLevelGetBufferHostSize\_8s32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8s32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.33 NppStatus nppiValidNormLevelGetBufferHostSize\_8u32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.34 NppStatus nppiValidNormLevelGetBufferHostSize\_8u32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).



**7.126.2.35 NppStatus nppiValidNormLevelGetBufferHostSize\_8u32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.36 NppStatus nppiValidNormLevelGetBufferHostSize\_8u32f\_C4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u32f\\_C4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.37 NppStatus nppiValidNormLevelGetBufferHostSize\_8u\_AC4RSfs (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u\\_AC4RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.38 NppStatus nppiValidNormLevelGetBufferHostSize\_8u\_C1RSfs (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u\\_C1RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.39 NppStatus nppiValidNormLevelGetBufferHostSize\_8u\_C3RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u\\_C3RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.126.2.40 NppStatus nppiValidNormLevelGetBufferHostSize\_8u\_C4RSfs (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiCrossCorrValid\\_NormLevel\\_8u\\_C4RSfs](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.127 Image Quality Index

Primitives for computing the image quality index of two images.

### QualityIndex

Given two images  $M$  and  $N$  (both  $W \times H$ ), the mathematical formula to calculate the image quality index  $Q$  between them is expressed as:

$$Q = \frac{4\sigma_{MN}\tilde{M}\tilde{N}}{[(\tilde{M}^2) + (\tilde{N}^2)][(\sigma_M)^2 + (\sigma_N)^2]}$$

where

$$\tilde{M} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} M(j, i)$$

$$\tilde{N} = \frac{1}{W \cdot H} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} N(j, i)$$

$$\sigma_M = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}]^2}$$

$$\sigma_N = \sqrt{\frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [N(j, i) - \tilde{N}]^2}$$

$$\sigma_{MN} = \frac{1}{W \cdot H - 1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} [M(j, i) - \tilde{M}][N(j, i) - \tilde{N}]$$

The functions require additional scratch buffer for computations.

- [NppStatus nppiQualityIndex\\_8u32f\\_C1R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)

*One-channel 8-bit unsigned image QualityIndex.*

- [NppStatus nppiQualityIndex\\_16u32f\\_C1R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)

*One-channel 16-bit unsigned image QualityIndex.*

- [NppStatus nppiQualityIndex\\_32f\\_C1R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)

*One-channel 32-bit floating point image QualityIndex.*

- [NppStatus nppiQualityIndex\\_8u32f\\_C3R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)

*Three-channel 8-bit unsigned image QualityIndex.*

- [NppStatus nppiQualityIndex\\_16u32f\\_C3R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oRoiSize, [Npp32f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)

*Three-channel 16-bit unsigned image QualityIndex.*

- **NppStatus** **nppiQualityIndex\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** \*pDst, **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image QualityIndex.*
- **NppStatus** **nppiQualityIndex\_8u32f\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** \*pDst, **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image QualityIndex.*
- **NppStatus** **nppiQualityIndex\_16u32f\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** \*pDst, **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image QualityIndex.*
- **NppStatus** **nppiQualityIndex\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oRoiSize, **Npp32f** \*pDst, **Npp8u** \*pDeviceBuffer)  
*Four-channel 32-bit floating point image QualityIndex.*

## QualityIndexGetBufferHostSize

Companion primitives for computing the device buffer size (in bytes) required by the QualityIndex primitives.

- **NppStatus** **nppiQualityIndexGetBufferHostSize\_8u32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size (in bytes) for **nppiQualityIndex\_8u32f\_C1R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize\_16u32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size (in bytes) for **nppiQualityIndex\_16u32f\_C1R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize\_32f\_C1R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size (in bytes) for **nppiQualityIndex\_32f\_C1R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize\_8u32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size (in bytes) for **nppiQualityIndex\_8u32f\_C3R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize\_16u32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size (in bytes) for **nppiQualityIndex\_16u32f\_C3R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize\_32f\_C3R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size (in bytes) for **nppiQualityIndex\_32f\_C3R**.*
- **NppStatus** **nppiQualityIndexGetBufferHostSize\_8u32f\_AC4R** (**NppiSize** oSizeROI, int \*hpBufferSize)  
*Buffer size (in bytes) for **nppiQualityIndex\_8u32f\_AC4R**.*

- **NppStatus** [nppiQualityIndexGetBufferHostSize\\_16u32f\\_AC4R](#) (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size (in bytes) for [nppiQualityIndex\\_16u32f\\_AC4R](#).
- **NppStatus** [nppiQualityIndexGetBufferHostSize\\_32f\\_AC4R](#) (**NppiSize** **oSizeROI**, **int** **\*hpBufferSize**)  
Buffer size (in bytes) for [nppiQualityIndex\\_32f\\_AC4R](#).

### 7.127.1 Detailed Description

Primitives for computing the image quality index of two images.

### 7.127.2 Function Documentation

**7.127.2.1** **NppStatus** [nppiQualityIndex\\_16u32f\\_AC4R](#) (**const Npp16u** \* *pSrc1*, **int** *nSrc1Step*, **const Npp16u** \* *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oRoiSize*, **Npp32f** \* *pDst*, **Npp8u** \* *pDeviceBuffer*)

Four-channel 16-bit unsigned image QualityIndex.

#### Parameters:

- pSrc1* [Source-Image Pointer](#).
- nSrc1Step* [Source-Image Line Step](#).
- pSrc2* [Source-Image Pointer](#).
- nSrc2Step* [Source-Image Line Step](#).
- oRoiSize* [Region-of-Interest \(ROI\)](#).
- pDst* [Pointer to the quality index](#).
- pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_16u32f\\_AC4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or [NPP\\_QUALITY\\_INDEX\\_ERROR](#) if pixels of either image are constant numberse.

**7.127.2.2** **NppStatus** [nppiQualityIndex\\_16u32f\\_C1R](#) (**const Npp16u** \* *pSrc1*, **int** *nSrc1Step*, **const Npp16u** \* *pSrc2*, **int** *nSrc2Step*, **NppiSize** *oRoiSize*, **Npp32f** \* *pDst*, **Npp8u** \* *pDeviceBuffer*)

One-channel 16-bit unsigned image QualityIndex.

#### Parameters:

- pSrc1* [Source-Image Pointer](#).
- nSrc1Step* [Source-Image Line Step](#).
- pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oRoiSize* Region-of-Interest (ROI).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_16u32f\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_QUALITY\_INDEX\_ERROR if pixels of either image are constant numberse.

**7.127.2.3** `NppStatus nppiQualityIndex_16u32f_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image QualityIndex.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oRoiSize* Region-of-Interest (ROI).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_16u32f\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_QUALITY\_INDEX\_ERROR if pixels of either image are constant numberse.

**7.127.2.4** `NppStatus nppiQualityIndex_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image QualityIndex.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oRoiSize* Region-of-Interest (ROI).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_QUALITY\_INDEX\_ERROR if pixels of either image are constant numberse.

**7.127.2.5 NppStatus nppiQualityIndex\_32f\_C1R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f \* pDst, Npp8u \* pDeviceBuffer)**

One-channel 32-bit floating point image QualityIndex.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_QUALITY\_INDEX\_ERROR if pixels of either image are constant numberse.

**7.127.2.6 NppStatus nppiQualityIndex\_32f\_C3R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f \* pDst, Npp8u \* pDeviceBuffer)**

Three-channel 32-bit floating point image QualityIndex.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oRoiSize* [Region-of-Interest \(ROI\)](#).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_QUALITY\_INDEX\_ERROR if pixels of either image are constant numberse.

**7.127.2.7** `NppStatus nppiQualityIndex_8u32f_AC4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image QualityIndex.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oRoiSize* Region-of-Interest (ROI).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_8u32f\\_AC4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

**7.127.2.8** `NppStatus nppiQualityIndex_8u32f_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image QualityIndex.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oRoiSize* Region-of-Interest (ROI).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_8u32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_QUALITY_INDEX_ERROR` if pixels of either image are constant numberse.

**7.127.2.9** `NppStatus nppiQualityIndex_8u32f_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oRoiSize, Npp32f * pDst, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image QualityIndex.



**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oRoiSize* Region-of-Interest (ROI).

*pDst* Pointer to the quality index.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiQualityIndexGetBufferHostSize\\_8u32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_QUALITY\_INDEX\_ERROR if pixels of either image are constant numberse.

#### 7.127.2.10 NppStatus nppiQualityIndexGetBufferHostSize\_16u32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex\\_16u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

#### 7.127.2.11 NppStatus nppiQualityIndexGetBufferHostSize\_16u32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)

Buffer size (in bytes) for [nppiQualityIndex\\_16u32f\\_C1R](#).

**Parameters:**

*oSizeROI* Region-of-Interest (ROI).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if hpBufferSize is 0 (NULL), [ROI Related Error Codes](#).

**7.127.2.12 NppStatus nppiQualityIndexGetBufferHostSize\_16u32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiQualityIndex\\_16u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.127.2.13 NppStatus nppiQualityIndexGetBufferHostSize\_32f\_AC4R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiQualityIndex\\_32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.127.2.14 NppStatus nppiQualityIndexGetBufferHostSize\_32f\_C1R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiQualityIndex\\_32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.127.2.15 NppStatus nppiQualityIndexGetBufferHostSize\_32f\_C3R (NppiSize *oSizeROI*, int \* *hpBufferSize*)**

Buffer size (in bytes) for [nppiQualityIndex\\_32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.127.2.16 NppStatus nppiQualityIndexGetBufferHostSize\_8u32f\_AC4R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiQualityIndex\\_8u32f\\_AC4R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.127.2.17 NppStatus nppiQualityIndexGetBufferHostSize\_8u32f\_C1R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiQualityIndex\\_8u32f\\_C1R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

**7.127.2.18 NppStatus nppiQualityIndexGetBufferHostSize\_8u32f\_C3R (NppiSize oSizeROI, int \* hpBufferSize)**

Buffer size (in bytes) for [nppiQualityIndex\\_8u32f\\_C3R](#).

**Parameters:**

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_NULL\_POINTER\_ERROR if *hpBufferSize* is 0 (NULL), [ROI Related Error Codes](#).

## 7.128 MaximumError

Primitives for computing the maximum error between two images.

### Functions

- **NppStatus nppiMaximumError\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image Maximum\_Error.*
- **NppStatus nppiMaximumError\_8s\_C1R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit signed image Maximum\_Error.*
- **NppStatus nppiMaximumError\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image Maximum\_Error.*
- **NppStatus nppiMaximumError\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image Maximum\_Error.*
- **NppStatus nppiMaximumError\_16sc\_C1R** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed complex image Maximum\_Error.*
- **NppStatus nppiMaximumError\_32u\_C1R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit unsigned image Maximum\_Error.*
- **NppStatus nppiMaximumError\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed image Maximum\_Error.*
- **NppStatus nppiMaximumError\_32sc\_C1R** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed complex image Maximum\_Error.*
- **NppStatus nppiMaximumError\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image Maximum\_Error.*
- **NppStatus nppiMaximumError\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point complex image Maximum\_Error.*
- **NppStatus nppiMaximumError\_64f\_C1R** (const **Npp64f** \*pSrc1, int nSrc1Step, const **Npp64f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 64-bit floating point image Maximum\_Error.*

- `NppStatus nppiMaximumError_8u_C2R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 8-bit unsigned image Maximum\_Error.*

- `NppStatus nppiMaximumError_8s_C2R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 8-bit signed image Maximum\_Error.*

- `NppStatus nppiMaximumError_16u_C2R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 16-bit unsigned image Maximum\_Error.*

- `NppStatus nppiMaximumError_16s_C2R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 16-bit signed image Maximum\_Error.*

- `NppStatus nppiMaximumError_16sc_C2R` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 16-bit signed complex image Maximum\_Error.*

- `NppStatus nppiMaximumError_32u_C2R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 32-bit unsigned image Maximum\_Error.*

- `NppStatus nppiMaximumError_32s_C2R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 32-bit signed image Maximum\_Error.*

- `NppStatus nppiMaximumError_32sc_C2R` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 32-bit signed complex image Maximum\_Error.*

- `NppStatus nppiMaximumError_32f_C2R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 32-bit floating point image Maximum\_Error.*

- `NppStatus nppiMaximumError_32fc_C2R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 32-bit floating point complex image Maximum\_Error.*

- `NppStatus nppiMaximumError_64f_C2R` (const `Npp64f` \*pSrc1, int nSrc1Step, const `Npp64f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Two-channel 64-bit floating point image Maximum\_Error.*

- `NppStatus nppiMaximumError_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit unsigned image Maximum\_Error.*

- `NppStatus nppiMaximumError_8s_C3R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Three-channel 8-bit signed image Maximum\_Error.*

- [NppStatus nppiMaximumError\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_16s\\_C3R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit signed image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_16sc\\_C3R](#) (const [Npp16sc](#) \*pSrc1, int nSrc1Step, const [Npp16sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit signed complex image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_32u\\_C3R](#) (const [Npp32u](#) \*pSrc1, int nSrc1Step, const [Npp32u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit unsigned image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_32s\\_C3R](#) (const [Npp32s](#) \*pSrc1, int nSrc1Step, const [Npp32s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit signed image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_32sc\\_C3R](#) (const [Npp32sc](#) \*pSrc1, int nSrc1Step, const [Npp32sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit signed complex image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit floating point image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_32fc\\_C3R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit floating point complex image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_64f\\_C3R](#) (const [Npp64f](#) \*pSrc1, int nSrc1Step, const [Npp64f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 64-bit floating point image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_8s\\_C4R](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit signed image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image Maximum\_Error.*
- [NppStatus nppiMaximumError\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)

*Four-channel 16-bit signed image Maximum\_Error.*

- `NppStatus nppiMaximumError_16sc_C4R` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit signed complex image Maximum\_Error.*

- `NppStatus nppiMaximumError_32u_C4R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit unsigned image Maximum\_Error.*

- `NppStatus nppiMaximumError_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit signed image Maximum\_Error.*

- `NppStatus nppiMaximumError_32sc_C4R` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit signed complex image Maximum\_Error.*

- `NppStatus nppiMaximumError_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image Maximum\_Error.*

- `NppStatus nppiMaximumError_32fc_C4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point complex image Maximum\_Error.*

- `NppStatus nppiMaximumError_64f_C4R` (const `Npp64f` \*pSrc1, int nSrc1Step, const `Npp64f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 64-bit floating point image Maximum\_Error.*

### 7.128.1 Detailed Description

Primitives for computing the maximum error between two images.

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the maximum error is defined as the largest absolute difference between pixels of two images. If the image is in complex format, the absolute value of the complex number is provided.

### 7.128.2 Function Documentation

#### 7.128.2.1 `NppStatus nppiMaximumError_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

One-channel 16-bit signed image Maximum\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `npapiMaximumErrorGetBufferHostSize_16s_C1R` to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.2** `NppStatus npapiMaximumError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `npapiMaximumErrorGetBufferHostSize_16s_C2R` to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.3** `NppStatus npapiMaximumError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use `npapiMaximumErrorGetBufferHostSize_16s_C3R` to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.128.2.4** `NppStatus nppiMaximumError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.5** `NppStatus nppiMaximumError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.6** `NppStatus nppiMaximumError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.7** `NppStatus nppiMaximumError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.8** `NppStatus nppiMaximumError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.9** `NppStatus nppiMaximumError_16u_C1R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

One-channel 16-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.10** `NppStatus nppiMaximumError_16u_C2R (const Npp16u *pSrc1, int nSrc1Step, const Npp16u *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Two-channel 16-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.11** `NppStatus nppiMaximumError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.12** `NppStatus nppiMaximumError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.13** `NppStatus nppiMaximumError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.14** `NppStatus nppiMaximumError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.15** `NppStatus nppiMaximumError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.16** `NppStatus nppiMaximumError_32f_C4R (const Npp32f *pSrc1, int nSrc1Step, const Npp32f *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Four-channel 32-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.17** `NppStatus nppiMaximumError_32fc_C1R (const Npp32fc *pSrc1, int nSrc1Step, const Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

One-channel 32-bit floating point complex image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.18** `NppStatus nppiMaximumError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.19** `NppStatus nppiMaximumError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.20** `NppStatus nppiMaximumError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.128.2.21** `NppStatus nppiMaximumError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.22** `NppStatus nppiMaximumError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image Maximum\_Error.



**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.23** `NppStatus nppiMaximumError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.24** `NppStatus nppiMaximumError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.25** `NppStatus nppiMaximumError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.26** `NppStatus nppiMaximumError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.27** `NppStatus nppiMaximumError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.28** `NppStatus nppiMaximumError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.29** `NppStatus nppiMaximumError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.30** `NppStatus nppiMaximumError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.31** `NppStatus nppiMaximumError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.32** `NppStatus nppiMaximumError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.33** `NppStatus nppiMaximumError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.128.2.34** `NppStatus nppiMaximumError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.128.2.35** `NppStatus nppiMaximumError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.128.2.36** `NppStatus nppiMaximumError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.128.2.37** `NppStatus nppiMaximumError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.38** `NppStatus nppiMaximumError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.39** `NppStatus nppiMaximumError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image Maximum\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.40** `NppStatus nppiMaximumError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image Maximum\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.128.2.41** `NppStatus nppiMaximumError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.42** `NppStatus nppiMaximumError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.43** `NppStatus nppiMaximumError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.128.2.44** `NppStatus nppiMaximumError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Maximum\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

## 7.129 AverageError

Primitives for computing the average error between two images.

### Functions

- **NppStatus nppiAverageError\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image Average\_Error.*
- **NppStatus nppiAverageError\_8s\_C1R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit signed image Average\_Error.*
- **NppStatus nppiAverageError\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image Average\_Error.*
- **NppStatus nppiAverageError\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image Average\_Error.*
- **NppStatus nppiAverageError\_16sc\_C1R** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed complex image Average\_Error.*
- **NppStatus nppiAverageError\_32u\_C1R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit unsigned image Average\_Error.*
- **NppStatus nppiAverageError\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed image Average\_Error.*
- **NppStatus nppiAverageError\_32sc\_C1R** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed complex image Average\_Error.*
- **NppStatus nppiAverageError\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image Average\_Error.*
- **NppStatus nppiAverageError\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point complex image Average\_Error.*
- **NppStatus nppiAverageError\_64f\_C1R** (const **Npp64f** \*pSrc1, int nSrc1Step, const **Npp64f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 64-bit floating point image Average\_Error.*

- [NppStatus nppiAverageError\\_8u\\_C2R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 8-bit unsigned image Average\_Error.*
- [NppStatus nppiAverageError\\_8s\\_C2R](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 8-bit signed image Average\_Error.*
- [NppStatus nppiAverageError\\_16u\\_C2R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 16-bit unsigned image Average\_Error.*
- [NppStatus nppiAverageError\\_16s\\_C2R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 16-bit signed image Average\_Error.*
- [NppStatus nppiAverageError\\_16sc\\_C2R](#) (const [Npp16sc](#) \*pSrc1, int nSrc1Step, const [Npp16sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 16-bit signed complex image Average\_Error.*
- [NppStatus nppiAverageError\\_32u\\_C2R](#) (const [Npp32u](#) \*pSrc1, int nSrc1Step, const [Npp32u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 32-bit unsigned image Average\_Error.*
- [NppStatus nppiAverageError\\_32s\\_C2R](#) (const [Npp32s](#) \*pSrc1, int nSrc1Step, const [Npp32s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 32-bit signed image Average\_Error.*
- [NppStatus nppiAverageError\\_32sc\\_C2R](#) (const [Npp32sc](#) \*pSrc1, int nSrc1Step, const [Npp32sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 32-bit signed complex image Average\_Error.*
- [NppStatus nppiAverageError\\_32f\\_C2R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 32-bit floating point image Average\_Error.*
- [NppStatus nppiAverageError\\_32fc\\_C2R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 32-bit floating point complex image Average\_Error.*
- [NppStatus nppiAverageError\\_64f\\_C2R](#) (const [Npp64f](#) \*pSrc1, int nSrc1Step, const [Npp64f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Two-channel 64-bit floating point image Average\_Error.*
- [NppStatus nppiAverageError\\_8u\\_C3R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image Average\_Error.*
- [NppStatus nppiAverageError\\_8s\\_C3R](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 8-bit signed image Average\_Error.*

- **NppStatus nppiAverageError\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image Average\_Error.*
- **NppStatus nppiAverageError\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed image Average\_Error.*
- **NppStatus nppiAverageError\_16sc\_C3R** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 16-bit signed complex image Average\_Error.*
- **NppStatus nppiAverageError\_32u\_C3R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit unsigned image Average\_Error.*
- **NppStatus nppiAverageError\_32s\_C3R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit signed image Average\_Error.*
- **NppStatus nppiAverageError\_32sc\_C3R** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit signed complex image Average\_Error.*
- **NppStatus nppiAverageError\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point image Average\_Error.*
- **NppStatus nppiAverageError\_32fc\_C3R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 32-bit floating point complex image Average\_Error.*
- **NppStatus nppiAverageError\_64f\_C3R** (const **Npp64f** \*pSrc1, int nSrc1Step, const **Npp64f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 64-bit floating point image Average\_Error.*
- **NppStatus nppiAverageError\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image Average\_Error.*
- **NppStatus nppiAverageError\_8s\_C4R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Four-channel 8-bit signed image Average\_Error.*
- **NppStatus nppiAverageError\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image Average\_Error.*
- **NppStatus nppiAverageError\_16s\_C4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit signed image Average\_Error.*

- **NppStatus** **nppiAverageError\_16sc\_C4R** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit signed complex image Average\_Error.*

- **NppStatus** **nppiAverageError\_32u\_C4R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit unsigned image Average\_Error.*

- **NppStatus** **nppiAverageError\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit signed image Average\_Error.*

- **NppStatus** **nppiAverageError\_32sc\_C4R** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit signed complex image Average\_Error.*

- **NppStatus** **nppiAverageError\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit floating point image Average\_Error.*

- **NppStatus** **nppiAverageError\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit floating point complex image Average\_Error.*

- **NppStatus** **nppiAverageError\_64f\_C4R** (const **Npp64f** \*pSrc1, int nSrc1Step, const **Npp64f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 64-bit floating point image Average\_Error.*

## 7.129.1 Detailed Description

Primitives for computing the average error between two images.

Given two images *pSrc1* and *pSrc2* both with width *W* and height *H*, the average error is defined as:

$$AverageError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} |pSrc1(j, i) - pSrc2(j, i)|$$

where *N* stands for the number of channels. If the image is in complex format, the absolute value is used for computation.

## 7.129.2 Function Documentation

- 7.129.2.1** **NppStatus nppiAverageError\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

One-channel 16-bit signed image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.2** `NppStatus nppiAverageError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.3** `NppStatus nppiAverageError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.4** `NppStatus nppiAverageError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.5** `NppStatus nppiAverageError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.129.2.6** `NppStatus nppiAverageError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.7** `NppStatus nppiAverageError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.8** `NppStatus nppiAverageError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.9** `NppStatus nppiAverageError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.10** `NppStatus nppiAverageError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.11** `NppStatus nppiAverageError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.12** `NppStatus nppiAverageError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.13** `NppStatus nppiAverageError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.129.2.14** `NppStatus nppiAverageError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.129.2.15** `NppStatus nppiAverageError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.16** `NppStatus nppiAverageError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.17** `NppStatus nppiAverageError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.18** `NppStatus nppiAverageError_32fc_C2R (const Npp32fc *pSrc1, int nSrc1Step, const Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Two-channel 32-bit floating point complex image Average\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.19** `NppStatus nppiAverageError_32fc_C3R (const Npp32fc *pSrc1, int nSrc1Step, const Npp32fc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Three-channel 32-bit floating point complex image Average\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.20** `NppStatus nppiAverageError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.21** `NppStatus nppiAverageError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.22** `NppStatus nppiAverageError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.23** `NppStatus nppiAverageError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.24** `NppStatus nppiAverageError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image Average\_Error.



**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.25** `NppStatus nppiAverageError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.26** `NppStatus nppiAverageError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.27** `NppStatus nppiAverageError_32sc_C3R (const Npp32sc *pSrc1, int nSrc1Step, const Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Three-channel 32-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.28** `NppStatus nppiAverageError_32sc_C4R (const Npp32sc *pSrc1, int nSrc1Step, const Npp32sc *pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f *pError, Npp8u *pDeviceBuffer)`

Four-channel 32-bit signed complex image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.29** `NppStatus nppiAverageError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.30** `NppStatus nppiAverageError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.31** `NppStatus nppiAverageError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.32** `NppStatus nppiAverageError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.33** `NppStatus nppiAverageError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.34** `NppStatus nppiAverageError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.129.2.35** `NppStatus nppiAverageError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.129.2.36** `NppStatus nppiAverageError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [npipiAverageErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.129.2.37** `NppStatus nppiAverageError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [npipiAverageErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.38** `NppStatus nppiAverageError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.39** `NppStatus nppiAverageError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image Average\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.40** `NppStatus nppiAverageError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.41** `NppStatus nppiAverageError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.42** `NppStatus nppiAverageError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).



*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.43** `NppStatus nppiAverageError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.129.2.44** `NppStatus nppiAverageError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image Average\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

## 7.130 MaximumRelativeError

Primitives for computing the maximum relative error between two images.

### Functions

- `NppStatus nppiMaximumRelativeError_8u_C1R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 8-bit unsigned image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_8s_C1R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 8-bit signed image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_16u_C1R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit unsigned image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit signed image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_16sc_C1R` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 16-bit signed complex image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32u_C1R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit unsigned image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32s_C1R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit signed image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32sc_C1R` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit signed complex image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32f_C1R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit floating point image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32fc_C1R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 32-bit floating point complex image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_64f_C1R` (const `Npp64f` \*pSrc1, int nSrc1Step, const `Npp64f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*One-channel 64-bit floating point image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_8u_C2R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 8-bit unsigned image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_8s_C2R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 8-bit signed image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_16u_C2R` (const `Npp16u` \*pSrc1, int nSrc1Step, const `Npp16u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 16-bit unsigned image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_16s_C2R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 16-bit signed image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_16sc_C2R` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 16-bit signed complex image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32u_C2R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 32-bit unsigned image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32s_C2R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 32-bit signed image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32sc_C2R` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 32-bit signed complex image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32f_C2R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 32-bit floating point image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_32fc_C2R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 32-bit floating point complex image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_64f_C2R` (const `Npp64f` \*pSrc1, int nSrc1Step, const `Npp64f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Two-channel 64-bit floating point image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_8u_C3R` (const `Npp8u` \*pSrc1, int nSrc1Step, const `Npp8u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image MaximumRelative\_Error.*
- `NppStatus nppiMaximumRelativeError_8s_C3R` (const `Npp8s` \*pSrc1, int nSrc1Step, const `Npp8s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)  
*Three-channel 8-bit signed image MaximumRelative\_Error.*

- [NppStatus nppiMaximumRelativeError\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_16s\\_C3R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit signed image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_16sc\\_C3R](#) (const [Npp16sc](#) \*pSrc1, int nSrc1Step, const [Npp16sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit signed complex image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_32u\\_C3R](#) (const [Npp32u](#) \*pSrc1, int nSrc1Step, const [Npp32u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_32s\\_C3R](#) (const [Npp32s](#) \*pSrc1, int nSrc1Step, const [Npp32s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit signed image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_32sc\\_C3R](#) (const [Npp32sc](#) \*pSrc1, int nSrc1Step, const [Npp32sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit signed complex image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit floating point image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_32fc\\_C3R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit floating point complex image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_64f\\_C3R](#) (const [Npp64f](#) \*pSrc1, int nSrc1Step, const [Npp64f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 64-bit floating point image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_8s\\_C4R](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit signed image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiMaximumRelativeError\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)

*Four-channel 16-bit signed image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_16sc_C4R` (const `Npp16sc` \*pSrc1, int nSrc1Step, const `Npp16sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 16-bit signed complex image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_32u_C4R` (const `Npp32u` \*pSrc1, int nSrc1Step, const `Npp32u` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit unsigned image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_32s_C4R` (const `Npp32s` \*pSrc1, int nSrc1Step, const `Npp32s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit signed image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_32sc_C4R` (const `Npp32sc` \*pSrc1, int nSrc1Step, const `Npp32sc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit signed complex image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_32f_C4R` (const `Npp32f` \*pSrc1, int nSrc1Step, const `Npp32f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_32fc_C4R` (const `Npp32fc` \*pSrc1, int nSrc1Step, const `Npp32fc` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 32-bit floating point complex image MaximumRelative\_Error.*

- `NppStatus nppiMaximumRelativeError_64f_C4R` (const `Npp64f` \*pSrc1, int nSrc1Step, const `Npp64f` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

*Four-channel 64-bit floating point image MaximumRelative\_Error.*

### 7.130.1 Detailed Description

Primitives for computing the maximum relative error between two images.

Given two images  $pSrc1$  and  $pSrc2$  both with width  $W$  and height  $H$ , the maximum relative error is defined as:

$$MaximumRelativeError = \max \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

If the image is in complex format, the absolute value is used for computation. For multiple channels, the maximum relative error of all the channels is returned.

### 7.130.2 Function Documentation

- 7.130.2.1** `NppStatus nppiMaximumRelativeError_16s_C1R` (const `Npp16s` \*pSrc1, int nSrc1Step, const `Npp16s` \*pSrc2, int nSrc2Step, `NppiSize` oSizeROI, `Npp64f` \*pError, `Npp8u` \*pDeviceBuffer)

One-channel 16-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.2** `NppStatus nppiMaximumRelativeError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.3** `NppStatus nppiMaximumRelativeError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.4** `NppStatus nppiMaximumRelativeError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.5** `NppStatus nppiMaximumRelativeError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.6** `NppStatus nppiMaximumRelativeError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.7** `NppStatus nppiMaximumRelativeError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.130.2.8** `NppStatus nppiMaximumRelativeError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.9** `NppStatus nppiMaximumRelativeError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.10** `NppStatus nppiMaximumRelativeError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.11** `NppStatus nppiMaximumRelativeError_16u_C3R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.12** `NppStatus nppiMaximumRelativeError_16u_C4R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.13** `NppStatus nppiMaximumRelativeError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.130.2.14** `NppStatus nppiMaximumRelativeError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.15** `NppStatus nppiMaximumRelativeError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.16** `NppStatus nppiMaximumRelativeError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.17** `NppStatus nppiMaximumRelativeError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.18** `NppStatus nppiMaximumRelativeError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.19** `NppStatus nppiMaximumRelativeError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.130.2.20** `NppStatus nppiMaximumRelativeError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.130.2.21** `NppStatus nppiMaximumRelativeError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.22** `NppStatus nppiMaximumRelativeError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.23** `NppStatus nppiMaximumRelativeError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.24** `NppStatus nppiMaximumRelativeError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.25** `NppStatus nppiMaximumRelativeError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.



*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.26** `NppStatus nppiMaximumRelativeError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.27** `NppStatus nppiMaximumRelativeError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.28** `NppStatus nppiMaximumRelativeError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.29** `NppStatus nppiMaximumRelativeError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.30** `NppStatus nppiMaximumRelativeError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.31** `NppStatus nppiMaximumRelativeError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.32** `NppStatus nppiMaximumRelativeError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.33** `NppStatus nppiMaximumRelativeError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.130.2.34** `NppStatus nppiMaximumRelativeError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.35** `NppStatus nppiMaximumRelativeError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.36** `NppStatus nppiMaximumRelativeError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.130.2.37** `NppStatus nppiMaximumRelativeError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.38** `NppStatus nppiMaximumRelativeError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.39** `NppStatus nppiMaximumRelativeError_8s_C3R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.40** `NppStatus nppiMaximumRelativeError_8s_C4R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.41** `NppStatus nppiMaximumRelativeError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.42** `NppStatus nppiMaximumRelativeError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.43** `NppStatus nppiMaximumRelativeError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MaximumRelative\_Error.



**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.130.2.44** `NppStatus nppiMaximumRelativeError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiMaximumRelativeErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

## 7.131 AverageRelativeError

Primitives for computing the average relative error between two images.

### Functions

- **NppStatus** **nppiAverageRelativeError\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit unsigned image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_8s\_C1R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 8-bit signed image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit unsigned image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_16sc\_C1R** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 16-bit signed complex image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_32u\_C1R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit unsigned image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_32s\_C1R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_32sc\_C1R** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit signed complex image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_32fc\_C1R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 32-bit floating point complex image MaximumRelative\_Error.*
- **NppStatus** **nppiAverageRelativeError\_64f\_C1R** (const **Npp64f** \*pSrc1, int nSrc1Step, const **Npp64f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*One-channel 64-bit floating point image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_8u\_C2R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 8-bit unsigned image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_8s\_C2R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 8-bit signed image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_16u\_C2R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 16-bit unsigned image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_16s\_C2R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 16-bit signed image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_16sc\_C2R** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 16-bit signed complex image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_32u\_C2R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 32-bit unsigned image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_32s\_C2R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 32-bit signed image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_32sc\_C2R** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 32-bit signed complex image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_32f\_C2R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 32-bit floating point image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_32fc\_C2R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 32-bit floating point complex image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_64f\_C2R** (const **Npp64f** \*pSrc1, int nSrc1Step, const **Npp64f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Two-channel 64-bit floating point image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit unsigned image MaximumRelative\_Error.*
- **NppStatus nppiAverageRelativeError\_8s\_C3R** (const **Npp8s** \*pSrc1, int nSrc1Step, const **Npp8s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)  
*Three-channel 8-bit signed image MaximumRelative\_Error.*

- [NppStatus nppiAverageRelativeError\\_16u\\_C3R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_16s\\_C3R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit signed image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_16sc\\_C3R](#) (const [Npp16sc](#) \*pSrc1, int nSrc1Step, const [Npp16sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 16-bit signed complex image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_32u\\_C3R](#) (const [Npp32u](#) \*pSrc1, int nSrc1Step, const [Npp32u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_32s\\_C3R](#) (const [Npp32s](#) \*pSrc1, int nSrc1Step, const [Npp32s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit signed image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_32sc\\_C3R](#) (const [Npp32sc](#) \*pSrc1, int nSrc1Step, const [Npp32sc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit signed complex image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc1, int nSrc1Step, const [Npp32f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit floating point image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_32fc\\_C3R](#) (const [Npp32fc](#) \*pSrc1, int nSrc1Step, const [Npp32fc](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 32-bit floating point complex image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_64f\\_C3R](#) (const [Npp64f](#) \*pSrc1, int nSrc1Step, const [Npp64f](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Three-channel 64-bit floating point image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_8u\\_C4R](#) (const [Npp8u](#) \*pSrc1, int nSrc1Step, const [Npp8u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_8s\\_C4R](#) (const [Npp8s](#) \*pSrc1, int nSrc1Step, const [Npp8s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 8-bit signed image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_16u\\_C4R](#) (const [Npp16u](#) \*pSrc1, int nSrc1Step, const [Npp16u](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)  
*Four-channel 16-bit unsigned image MaximumRelative\_Error.*
- [NppStatus nppiAverageRelativeError\\_16s\\_C4R](#) (const [Npp16s](#) \*pSrc1, int nSrc1Step, const [Npp16s](#) \*pSrc2, int nSrc2Step, [NppiSize](#) oSizeROI, [Npp64f](#) \*pError, [Npp8u](#) \*pDeviceBuffer)

*Four-channel 16-bit signed image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_16sc\_C4R** (const **Npp16sc** \*pSrc1, int nSrc1Step, const **Npp16sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 16-bit signed complex image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_32u\_C4R** (const **Npp32u** \*pSrc1, int nSrc1Step, const **Npp32u** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit unsigned image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_32s\_C4R** (const **Npp32s** \*pSrc1, int nSrc1Step, const **Npp32s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit signed image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_32sc\_C4R** (const **Npp32sc** \*pSrc1, int nSrc1Step, const **Npp32sc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit signed complex image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit floating point image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_32fc\_C4R** (const **Npp32fc** \*pSrc1, int nSrc1Step, const **Npp32fc** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 32-bit floating point complex image MaximumRelative\_Error.*

- **NppStatus nppiAverageRelativeError\_64f\_C4R** (const **Npp64f** \*pSrc1, int nSrc1Step, const **Npp64f** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

*Four-channel 64-bit floating point image MaximumRelative\_Error.*

### 7.131.1 Detailed Description

Primitives for computing the average relative error between two images.

Given two images  $pSrc1$  and  $pSrc2$  both with width  $W$  and height  $H$ , the maximum relative error is defined as:

$$AverageRelativeError = \frac{1}{W \cdot H \cdot N} \sum_{n=0}^{N-1} \sum_{j=0}^{H-1} \sum_{i=0}^{W-1} \frac{|pSrc1(j, i) - pSrc2(j, i)|}{\max(|pSrc1(j, i)|, |pSrc2(j, i)|)}$$

where  $N$  is the number of channels. If the image is in complex format, the absolute value is used for computation.

### 7.131.2 Function Documentation

- 7.131.2.1 NppStatus nppiAverageRelativeError\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **NppiSize** oSizeROI, **Npp64f** \*pError, **Npp8u** \*pDeviceBuffer)

One-channel 16-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.2** `NppStatus nppiAverageRelativeError_16s_C2R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.3** `NppStatus nppiAverageRelativeError_16s_C3R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.4** `NppStatus nppiAverageRelativeError_16s_C4R (const Npp16s * pSrc1, int nSrc1Step, const Npp16s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.5** `NppStatus nppiAverageRelativeError_16sc_C1R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit signed complex image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* Source-Image Line Step.

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.6** `NppStatus nppiAverageRelativeError_16sc_C2R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.7** `NppStatus nppiAverageRelativeError_16sc_C3R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 16-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



**7.131.2.8** `NppStatus nppiAverageRelativeError_16sc_C4R (const Npp16sc * pSrc1, int nSrc1Step, const Npp16sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 16-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* [Pointer to the computed error \(absolute value\)](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.9** `NppStatus nppiAverageRelativeError_16u_C1R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* [Pointer to the computed error](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.10** `NppStatus nppiAverageRelativeError_16u_C2R (const Npp16u * pSrc1, int nSrc1Step, const Npp16u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.11 NppStatus nppiAverageRelativeError\_16u\_C3R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f \* *pError*, Npp8u \* *pDeviceBuffer*)

Three-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.12 NppStatus nppiAverageRelativeError\_16u\_C4R** (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, NppiSize *oSizeROI*, Npp64f \* *pError*, Npp8u \* *pDeviceBuffer*)

Four-channel 16-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.13** `NppStatus nppiAverageRelativeError_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.14** `NppStatus nppiAverageRelativeError_32f_C2R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.131.2.15** `NppStatus nppiAverageRelativeError_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.131.2.16** `NppStatus nppiAverageRelativeError_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.17** `NppStatus nppiAverageRelativeError_32fc_C1R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.18** `NppStatus nppiAverageRelativeError_32fc_C2R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#). Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.19** `NppStatus nppiAverageRelativeError_32fc_C3R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.20** `NppStatus nppiAverageRelativeError_32fc_C4R (const Npp32fc * pSrc1, int nSrc1Step, const Npp32fc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit floating point complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.21** `NppStatus nppiAverageRelativeError_32s_C1R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.22** `NppStatus nppiAverageRelativeError_32s_C2R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.23** `NppStatus nppiAverageRelativeError_32s_C3R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.24** `NppStatus nppiAverageRelativeError_32s_C4R (const Npp32s * pSrc1, int nSrc1Step, const Npp32s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.25** `NppStatus nppiAverageRelativeError_32sc_C1R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.



*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C1R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.26** `NppStatus nppiAverageRelativeError_32sc_C2R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit signed complex image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C2R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.27** `NppStatus nppiAverageRelativeError_32sc_C3R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit signed complex image MaximumRelative\_Error.

#### Parameters:

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error (absolute value).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.28** `NppStatus nppiAverageRelativeError_32sc_C4R (const Npp32sc * pSrc1, int nSrc1Step, const Npp32sc * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit signed complex image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error (absolute value).  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16s\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.29** `NppStatus nppiAverageRelativeError_32u_C1R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.30** `NppStatus nppiAverageRelativeError_32u_C2R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.31** `NppStatus nppiAverageRelativeError_32u_C3R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.32** `NppStatus nppiAverageRelativeError_32u_C4R (const Npp32u * pSrc1, int nSrc1Step, const Npp32u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 32-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_16u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.33** `NppStatus nppiAverageRelativeError_64f_C1R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 64-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.34** `NppStatus nppiAverageRelativeError_64f_C2R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 64-bit floating point image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C2R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.131.2.35** `NppStatus nppiAverageRelativeError_64f_C3R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 64-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C3R](#) to compute the required size (in bytes).

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_EVEN\_STEP\_ERROR if an invalid floating-point image is specified.

**7.131.2.36** `NppStatus nppiAverageRelativeError_64f_C4R (const Npp64f * pSrc1, int nSrc1Step, const Npp64f * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 64-bit floating point image MaximumRelative\_Error.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_32f\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_EVEN_STEP_ERROR` if an invalid floating-point image is specified.

**7.131.2.37** `NppStatus nppiAverageRelativeError_8s_C1R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.38** `NppStatus nppiAverageRelativeError_8s_C2R (const Npp8s * pSrc1, int nSrc1Step, const Npp8s * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.131.2.39 NppStatus nppiAverageRelativeError\_8s\_C3R (const Npp8s \* pSrc1, int nSrc1Step, const Npp8s \* pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f \* pError, Npp8u \* pDeviceBuffer)

Three-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.131.2.40 NppStatus nppiAverageRelativeError\_8s\_C4R (const Npp8s \* pSrc1, int nSrc1Step, const Npp8s \* pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f \* pError, Npp8u \* pDeviceBuffer)

Four-channel 8-bit signed image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).  
*nSrc1Step* [Source-Image Line Step](#).  
*pSrc2* [Source-Image Pointer](#).  
*nSrc2Step* [Source-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*pError* Pointer to the computed error.  
*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
 Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.41** `NppStatus nppiAverageRelativeError_8u_C1R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

One-channel 8-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C1R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.42** `NppStatus nppiAverageRelativeError_8u_C2R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Two-channel 8-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C2R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.43** `NppStatus nppiAverageRelativeError_8u_C3R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Three-channel 8-bit unsigned image MaximumRelative\_Error.



**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C3R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.131.2.44** `NppStatus nppiAverageRelativeError_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, NppiSize oSizeROI, Npp64f * pError, Npp8u * pDeviceBuffer)`

Four-channel 8-bit unsigned image MaximumRelative\_Error.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*pError* Pointer to the computed error.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C4R](#) to compute the required size (in bytes).

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

## 7.132 Memory Management

Routines for allocating and deallocating pitched image storage.

### Functions

- void [nppiFree](#) (void \*pData)  
*Free method for any 2D allocated memory.*

### Image Memory Allocation

ImageAllocator methods for 2D arrays of data.

The allocators have width and height parameters to specify the size of the image data being allocated. They return a pointer to the newly created memory and return the numbers of bytes between successive lines.

If the memory allocation failed due to lack of free device memory or device memory fragmentation the routine returns 0.

All allocators return memory with line strides that are beneficial for performance. It is not mandatory to use these allocators. Any valid CUDA device-memory pointers can be used by the NPP primitives and there are no restrictions on line strides.

- [Npp8u \\* nppiMalloc\\_8u\\_C1](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*8-bit unsigned image memory allocator.*
- [Npp8u \\* nppiMalloc\\_8u\\_C2](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 8-bit unsigned image memory allocator.*
- [Npp8u \\* nppiMalloc\\_8u\\_C3](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 8-bit unsigned image memory allocator.*
- [Npp8u \\* nppiMalloc\\_8u\\_C4](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 8-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C1](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*16-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C2](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C3](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 16-bit unsigned image memory allocator.*
- [Npp16u \\* nppiMalloc\\_16u\\_C4](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit unsigned image memory allocator.*
- [Npp16s \\* nppiMalloc\\_16s\\_C1](#) (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*16-bit signed image memory allocator.*

- `Npp16s * nppiMalloc_16s_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit signed image memory allocator.*
- `Npp16s * nppiMalloc_16s_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit signed image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*1 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 16-bit signed complex image memory allocator.*
- `Npp16sc * nppiMalloc_16sc_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 16-bit signed complex image memory allocator.*
- `Npp32s * nppiMalloc_32s_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit signed image memory allocator.*
- `Npp32s * nppiMalloc_32s_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit signed image memory allocator.*
- `Npp32s * nppiMalloc_32s_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit signed image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit integer complex image memory allocator.*
- `Npp32sc * nppiMalloc_32sc_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit integer complex image memory allocator.*
- `Npp32f * nppiMalloc_32f_C1` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C2` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*2 channel 32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C3` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*3 channel 32-bit floating point image memory allocator.*
- `Npp32f * nppiMalloc_32f_C4` (int nWidthPixels, int nHeightPixels, int \*pStepBytes)  
*4 channel 32-bit floating point image memory allocator.*

- `Npp32fc * nppiMalloc_32fc_C1` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C2` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*2 channel 32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C3` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*3 channel 32-bit float complex image memory allocator.*
- `Npp32fc * nppiMalloc_32fc_C4` (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)  
*4 channel 32-bit float complex image memory allocator.*

### 7.132.1 Detailed Description

Routines for allocating and deallocating pitched image storage.

These methods are provided for convenience. They allocate memory that may contain additional padding bytes at the end of each line of pixels. Though padding is not necessary for any of the NPP image-processing primitives to work correctly, its absence may cause severe performance degradation compared to properly padded images.

### 7.132.2 Function Documentation

#### 7.132.2.1 void nppiFree (void \* *pData*)

Free method for any 2D allocated memory.

This method should be used to free memory allocated with any of the `nppiMalloc_<modifier>` methods.

##### Parameters:

*pData* A pointer to memory allocated using `nppiMalloc_<modifier>`.

#### 7.132.2.2 Npp16s\* nppiMalloc\_16s\_C1 (int *nWidthPixels*, int *nHeightPixels*, int *\*pStepBytes*)

16-bit signed image memory allocator.

##### Parameters:

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* Line Step.

##### Returns:

Pointer to new image data.

**7.132.2.3 Npp16s\* nppiMalloc\_16s\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.4 Npp16s\* nppiMalloc\_16s\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.5 Npp16sc\* nppiMalloc\_16sc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

1 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.6 Npp16sc\* nppiMalloc\_16sc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.7 Npp16sc\* nppiMalloc\_16sc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.8 Npp16sc\* nppiMalloc\_16sc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit signed complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.9 Npp16u\* nppiMalloc\_16u\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.10 Npp16u\* nppiMalloc\_16u\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.11 Npp16u\* nppiMalloc\_16u\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.12 Npp16u\* nppiMalloc\_16u\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 16-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.13 Npp32f\* nppiMalloc\_32f\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.14 Npp32f\* nppiMalloc\_32f\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.15 Npp32f\* nppiMalloc\_32f\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.16 Npp32f\* nppiMalloc\_32f\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit floating point image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.



**7.132.2.17 Npp32fc\* nppiMalloc\_32fc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.18 Npp32fc\* nppiMalloc\_32fc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.19 Npp32fc\* nppiMalloc\_32fc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.20 Npp32fc\* nppiMalloc\_32fc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit float complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.21 Npp32s\* nppiMalloc\_32s\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.22 Npp32s\* nppiMalloc\_32s\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.23 Npp32s\* nppiMalloc\_32s\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit signed image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.24 Npp32sc\* nppiMalloc\_32sc\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.25 Npp32sc\* nppiMalloc\_32sc\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.26 Npp32sc\* nppiMalloc\_32sc\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.27 Npp32sc\* nppiMalloc\_32sc\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 32-bit integer complex image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.28 Npp8u\* nppiMalloc\_8u\_C1 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.29 Npp8u\* nppiMalloc\_8u\_C2 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

2 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.30 Npp8u\* nppiMalloc\_8u\_C3 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

3 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

**7.132.2.31 Npp8u\* nppiMalloc\_8u\_C4 (int *nWidthPixels*, int *nHeightPixels*, int \* *pStepBytes*)**

4 channel 8-bit unsigned image memory allocator.

**Parameters:**

*nWidthPixels* Image width.

*nHeightPixels* Image height.

*pStepBytes* [Line Step](#).

**Returns:**

Pointer to new image data.

## 7.133 Threshold and Compare Operations

Methods for pixel-wise threshold and compare operations.

### Modules

- [Threshold Operations](#)

*Threshold image pixels.*

- [Compare Operations](#)

*Compare the pixels of two images and create a binary result image.*

### 7.133.1 Detailed Description

Methods for pixel-wise threshold and compare operations.

## 7.134 Threshold Operations

Threshold image pixels.

### Functions

- **NppStatus nppiThreshold\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 8-bit unsigned char threshold.*
- **NppStatus nppiThreshold\_8u\_C1IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 8-bit unsigned char in place threshold.*
- **NppStatus nppiThreshold\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit unsigned short threshold.*
- **NppStatus nppiThreshold\_16u\_C1IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit unsigned short in place threshold.*
- **NppStatus nppiThreshold\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit signed short threshold.*
- **NppStatus nppiThreshold\_16s\_C1IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit signed short in place threshold.*
- **NppStatus nppiThreshold\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 32-bit floating point threshold.*
- **NppStatus nppiThreshold\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold, **NppCmpOp** eComparisonOperation)  
*1 channel 32-bit floating point in place threshold.*
- **NppStatus nppiThreshold\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 8-bit unsigned char threshold.*
- **NppStatus nppiThreshold\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 8-bit unsigned char in place threshold.*
- **NppStatus nppiThreshold\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3], **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit unsigned short threshold.*

- `NppStatus nppiThreshold_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], `NppCmpOp` eComparisonOperation)  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*



- `NppStatus nppiThreshold_GT_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GT_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)  
*1 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GT_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GT_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GT_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_GT_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_GT_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_GT_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_GT_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GT_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GT_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GT_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GT_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

*3 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_GT_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

*3 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_GT_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

*3 channel 32-bit floating point threshold.*

- `NppStatus nppiThreshold_GT_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

*3 channel 32-bit floating point in place threshold.*

- `NppStatus nppiThreshold_GT_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])

*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])

*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])

*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])

*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

*4 channel 16-bit signed short image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])

*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

*4 channel 32-bit floating point image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GT_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])

*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_LT_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold)

*1 channel 8-bit unsigned char threshold.*

- **NppStatus nppiThreshold\_LT\_8u\_C1IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** nThreshold)  
*1 channel 8-bit unsigned char in place threshold.*
- **NppStatus nppiThreshold\_LT\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold)  
*1 channel 16-bit unsigned short threshold.*
- **NppStatus nppiThreshold\_LT\_16u\_C1IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** nThreshold)  
*1 channel 16-bit unsigned short in place threshold.*
- **NppStatus nppiThreshold\_LT\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold)  
*1 channel 16-bit signed short threshold.*
- **NppStatus nppiThreshold\_LT\_16s\_C1IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** nThreshold)  
*1 channel 16-bit signed short in place threshold.*
- **NppStatus nppiThreshold\_LT\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold)  
*1 channel 32-bit floating point threshold.*
- **NppStatus nppiThreshold\_LT\_32f\_C1IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** nThreshold)  
*1 channel 32-bit floating point in place threshold.*
- **NppStatus nppiThreshold\_LT\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3])  
*3 channel 8-bit unsigned char threshold.*
- **NppStatus nppiThreshold\_LT\_8u\_C3IR** (**Npp8u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp8u** rThresholds[3])  
*3 channel 8-bit unsigned char in place threshold.*
- **NppStatus nppiThreshold\_LT\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, **Npp16u** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3])  
*3 channel 16-bit unsigned short threshold.*
- **NppStatus nppiThreshold\_LT\_16u\_C3IR** (**Npp16u** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16u** rThresholds[3])  
*3 channel 16-bit unsigned short in place threshold.*
- **NppStatus nppiThreshold\_LT\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3])  
*3 channel 16-bit signed short threshold.*
- **NppStatus nppiThreshold\_LT\_16s\_C3IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3])  
*3 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_LT_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_LT_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_LT_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3])  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3])  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3])  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LT_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3])  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_Val_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_Val_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_Val_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_Val_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_Val_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_Val_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_Val_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue, `NppCmpOp` eComparisonOperation)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_Val_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_Val_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_Val_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_Val_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_Val_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_Val_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_Val_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_Val_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_Val_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_Val_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3], `NppCmpOp` eComparisonOperation)  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*

- `NppStatus nppiThreshold_GTVal_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GTVal_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)  
*1 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)  
*1 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)  
*1 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GTVal_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)  
*1 channel 16-bit signed short threshold.*
- `NppStatus nppiThreshold_GTVal_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)  
*1 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_GTVal_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)  
*1 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_GTVal_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)  
*1 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_GTVal_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*3 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_GTVal_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*3 channel 8-bit unsigned char in place threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*3 channel 16-bit unsigned short threshold.*
- `NppStatus nppiThreshold_GTVal_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*3 channel 16-bit unsigned short in place threshold.*
- `NppStatus nppiThreshold_GTVal_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])  
*3 channel 16-bit signed short threshold.*



- `NppStatus nppiThreshold_GTVa1_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])  
*3 channel 16-bit signed short in place threshold.*
- `NppStatus nppiThreshold_GTVa1_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_GTVa1_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_GTVa1_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVa1_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVa1_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVa1_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVa1_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVa1_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVa1_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_GTVa1_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVa1_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_LTVa1_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThreshold, const `Npp8u` nValue)



*1 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)

*1 channel 16-bit unsigned short threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThreshold, const `Npp16u` nValue)

*1 channel 16-bit unsigned short in place threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)

*1 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThreshold, const `Npp16s` nValue)

*1 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_LTVal_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)

*1 channel 32-bit floating point threshold.*

- `NppStatus nppiThreshold_LTVal_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThreshold, const `Npp32f` nValue)

*1 channel 32-bit floating point in place threshold.*

- `NppStatus nppiThreshold_LTVal_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])

*3 channel 8-bit unsigned char threshold.*

- `NppStatus nppiThreshold_LTVal_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])

*3 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])

*3 channel 16-bit unsigned short threshold.*

- `NppStatus nppiThreshold_LTVal_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])

*3 channel 16-bit unsigned short in place threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C3R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])

*3 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_LTVal_16s_C3IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])

*3 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_LTVal_32f_C3R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*3 channel 32-bit floating point threshold.*
- `NppStatus nppiThreshold_LTVal_32f_C3IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*3 channel 32-bit floating point in place threshold.*
- `NppStatus nppiThreshold_LTVal_8u_AC4R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_8u_AC4IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholds[3], const `Npp8u` rValues[3])  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16u_AC4R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16u_AC4IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholds[3], const `Npp16u` rValues[3])  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16s_AC4R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_16s_AC4IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` rThresholds[3], const `Npp16s` rValues[3])  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_32f_AC4R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*4 channel 32-bit floating point image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTVal_32f_AC4IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` rThresholds[3], const `Npp32f` rValues[3])  
*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*
- `NppStatus nppiThreshold_LTValGTVal_8u_C1R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` nThresholdLT, const `Npp8u` nValueLT, const `Npp8u` nThresholdGT, const `Npp8u` nValueGT)  
*1 channel 8-bit unsigned char threshold.*
- `NppStatus nppiThreshold_LTValGTVal_8u_C1IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` nThresholdLT, const `Npp8u` nValueLT, const `Npp8u` nThresholdGT, const `Npp8u` nValueGT)  
*1 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16u_C1R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` nThresholdLT, const `Npp16u` nValueLT, const `Npp16u` nThresholdGT, const `Npp16u` nValueGT)

*1 channel 16-bit unsigned short threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16u_C1IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` nThresholdLT, const `Npp16u` nValueLT, const `Npp16u` nThresholdGT, const `Npp16u` nValueGT)

*1 channel 16-bit unsigned short in place threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16s_C1R` (const `Npp16s` \*pSrc, int nSrcStep, `Npp16s` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16s` nThresholdLT, const `Npp16s` nValueLT, const `Npp16s` nThresholdGT, const `Npp16s` nValueGT)

*1 channel 16-bit signed short threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16s_C1IR` (`Npp16s` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16s` nThresholdLT, const `Npp16s` nValueLT, const `Npp16s` nThresholdGT, const `Npp16s` nValueGT)

*1 channel 16-bit signed short in place threshold.*

- `NppStatus nppiThreshold_LTValGTVal_32f_C1R` (const `Npp32f` \*pSrc, int nSrcStep, `Npp32f` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp32f` nThresholdLT, const `Npp32f` nValueLT, const `Npp32f` nThresholdGT, const `Npp32f` nValueGT)

*1 channel 32-bit floating point threshold.*

- `NppStatus nppiThreshold_LTValGTVal_32f_C1IR` (`Npp32f` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp32f` nThresholdLT, const `Npp32f` nValueLT, const `Npp32f` nThresholdGT, const `Npp32f` nValueGT)

*1 channel 32-bit floating point in place threshold.*

- `NppStatus nppiThreshold_LTValGTVal_8u_C3R` (const `Npp8u` \*pSrc, int nSrcStep, `Npp8u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])

*3 channel 8-bit unsigned char threshold.*

- `NppStatus nppiThreshold_LTValGTVal_8u_C3IR` (`Npp8u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp8u` rThresholdsLT[3], const `Npp8u` rValuesLT[3], const `Npp8u` rThresholdsGT[3], const `Npp8u` rValuesGT[3])

*3 channel 8-bit unsigned char in place threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16u_C3R` (const `Npp16u` \*pSrc, int nSrcStep, `Npp16u` \*pDst, int nDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])

*3 channel 16-bit unsigned short threshold.*

- `NppStatus nppiThreshold_LTValGTVal_16u_C3IR` (`Npp16u` \*pSrcDst, int nSrcDstStep, `NppiSize` oSizeROI, const `Npp16u` rThresholdsLT[3], const `Npp16u` rValuesLT[3], const `Npp16u` rThresholdsGT[3], const `Npp16u` rValuesGT[3])

*3 channel 16-bit unsigned short in place threshold.*

- [NppStatus nppiThreshold\\_LTValGTVal\\_16s\\_C3R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp16s](#) rThresholdsLT[3], const [Npp16s](#) rValuesLT[3], const [Npp16s](#) rThresholdsGT[3], const [Npp16s](#) rValuesGT[3])  
*3 channel 16-bit signed short threshold.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_16s\\_C3IR](#) ([Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const [Npp16s](#) rThresholdsLT[3], const [Npp16s](#) rValuesLT[3], const [Npp16s](#) rThresholdsGT[3], const [Npp16s](#) rValuesGT[3])  
*3 channel 16-bit signed short in place threshold.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_32f\\_C3R](#) (const [Npp32f](#) \*pSrc, int nSrcStep, [Npp32f](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp32f](#) rThresholdsLT[3], const [Npp32f](#) rValuesLT[3], const [Npp32f](#) rThresholdsGT[3], const [Npp32f](#) rValuesGT[3])  
*3 channel 32-bit floating point threshold.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_32f\\_C3IR](#) ([Npp32f](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const [Npp32f](#) rThresholdsLT[3], const [Npp32f](#) rValuesLT[3], const [Npp32f](#) rThresholdsGT[3], const [Npp32f](#) rValuesGT[3])  
*3 channel 32-bit floating point in place threshold.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_8u\\_AC4R](#) (const [Npp8u](#) \*pSrc, int nSrcStep, [Npp8u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) rThresholdsLT[3], const [Npp8u](#) rValuesLT[3], const [Npp8u](#) rThresholdsGT[3], const [Npp8u](#) rValuesGT[3])  
*4 channel 8-bit unsigned char image threshold, not affecting Alpha.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_8u\\_AC4IR](#) ([Npp8u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const [Npp8u](#) rThresholdsLT[3], const [Npp8u](#) rValuesLT[3], const [Npp8u](#) rThresholdsGT[3], const [Npp8u](#) rValuesGT[3])  
*4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_16u\\_AC4R](#) (const [Npp16u](#) \*pSrc, int nSrcStep, [Npp16u](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp16u](#) rThresholdsLT[3], const [Npp16u](#) rValuesLT[3], const [Npp16u](#) rThresholdsGT[3], const [Npp16u](#) rValuesGT[3])  
*4 channel 16-bit unsigned short image threshold, not affecting Alpha.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_16u\\_AC4IR](#) ([Npp16u](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const [Npp16u](#) rThresholdsLT[3], const [Npp16u](#) rValuesLT[3], const [Npp16u](#) rThresholdsGT[3], const [Npp16u](#) rValuesGT[3])  
*4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_16s\\_AC4R](#) (const [Npp16s](#) \*pSrc, int nSrcStep, [Npp16s](#) \*pDst, int nDstStep, [NppiSize](#) oSizeROI, const [Npp16s](#) rThresholdsLT[3], const [Npp16s](#) rValuesLT[3], const [Npp16s](#) rThresholdsGT[3], const [Npp16s](#) rValuesGT[3])  
*4 channel 16-bit signed short image threshold, not affecting Alpha.*
- [NppStatus nppiThreshold\\_LTValGTVal\\_16s\\_AC4IR](#) ([Npp16s](#) \*pSrcDst, int nSrcDstStep, [NppiSize](#) oSizeROI, const [Npp16s](#) rThresholdsLT[3], const [Npp16s](#) rValuesLT[3], const [Npp16s](#) rThresholdsGT[3], const [Npp16s](#) rValuesGT[3])  
*4 channel 16-bit signed short in place image threshold, not affecting Alpha.*

- **NppStatus** **nppiThreshold\_LTValGTVal\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, **Npp32f** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholdsLT[3], const **Npp32f** rValuesLT[3], const **Npp32f** rThresholdsGT[3], const **Npp32f** rValuesGT[3])

*4 channel 32-bit floating point image threshold, not affecting Alpha.*

- **NppStatus** **nppiThreshold\_LTValGTVal\_32f\_AC4IR** (**Npp32f** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp32f** rThresholdsLT[3], const **Npp32f** rValuesLT[3], const **Npp32f** rThresholdsGT[3], const **Npp32f** rValuesGT[3])

*4 channel 32-bit floating point in place image threshold, not affecting Alpha.*

### 7.134.1 Detailed Description

Threshold image pixels.

### 7.134.2 Function Documentation

#### 7.134.2.1 **NppStatus nppiThreshold\_16s\_AC4IR** (**Npp16s** \*pSrcDst, int nSrcDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

**pSrcDst** [In-Place Image Pointer](#).

**nSrcDstStep** [In-Place-Image Line Step](#).

**oSizeROI** [Region-of-Interest \(ROI\)](#).

**rThresholds** The threshold values, one per color channel.

**eComparisonOperation** The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.134.2.2 **NppStatus nppiThreshold\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, **Npp16s** \*pDst, int nDstStep, **NppiSize** oSizeROI, const **Npp16s** rThresholds[3], **NppCmpOp** eComparisonOperation)

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

**pSrc** [Source-Image Pointer](#).

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.3 NppStatus nppiThreshold\_16s\_C1IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, NppCmpOp eComparisonOperation)**

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.4 NppStatus nppiThreshold\_16s\_C1R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, NppCmpOp eComparisonOperation)**

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.5 NppStatus nppiThreshold\_16s\_C3IR (Npp16s \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], NppCmpOp eComparisonOperation)**

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.6 NppStatus nppiThreshold\_16s\_C3R (const Npp16s \* pSrc, int nSrcStep, Npp16s \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], NppCmpOp eComparisonOperation)**

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

#### 7.134.2.7 `NppStatus nppiThreshold_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (`sourcePixel.channel OP nThreshold`) is true, the channel value is set to `nThreshold`, otherwise it is set to `sourcePixel`.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.

#### 7.134.2.8 `NppStatus nppiThreshold_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (`sourcePixel.channel OP nThreshold`) is true, the channel value is set to `nThreshold`, otherwise it is set to `sourcePixel`.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: `NPP_CMP_LESS` and `NPP_CMP_GREATER`.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or `NPP_NOT_SUPPORTED_MODE_ERROR` if an invalid comparison operation type is specified.



### 7.134.2.9 **NppStatus nppiThreshold\_16u\_C1IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, NppCmpOp *eComparisonOperation*)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

### 7.134.2.10 **NppStatus nppiThreshold\_16u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, NppCmpOp *eComparisonOperation*)

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

### 7.134.2.11 **NppStatus nppiThreshold\_16u\_C3IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.134.2.12 **NppStatus nppiThreshold\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.134.2.13 **NppStatus nppiThreshold\_32f\_AC4IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], NppCmpOp *eComparisonOperation*)

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.14 NppStatus nppiThreshold\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)**

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.15 NppStatus nppiThreshold\_32f\_C1IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, NppCmpOp eComparisonOperation)**

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.16 NppStatus nppiThreshold\_32f\_C1R (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*, NppCmpOp *eComparisonOperation*)**

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.17 NppStatus nppiThreshold\_32f\_C3IR (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], NppCmpOp *eComparisonOperation*)**

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.18 NppStatus nppiThreshold\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], NppCmpOp eComparisonOperation)**

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.19 NppStatus nppiThreshold\_8u\_AC4IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], NppCmpOp eComparisonOperation)**

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.134.2.20 **NppStatus nppiThreshold\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.134.2.21 **NppStatus nppiThreshold\_8u\_C1IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, NppCmpOp *eComparisonOperation*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.22 NppStatus nppiThreshold\_8u\_C1R (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, NppCmpOp *eComparisonOperation*)**

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.23 NppStatus nppiThreshold\_8u\_C3IR (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], NppCmpOp *eComparisonOperation*)**

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.134.2.24 **NppStatus nppiThreshold\_8u\_C3R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], NppCmpOp *eComparisonOperation*)

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

#### 7.134.2.25 **NppStatus nppiThreshold\_GT\_16s\_AC4IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.26 **NppStatus nppiThreshold\_GT\_16s\_AC4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.



**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.27 **NppStatus nppiThreshold\_GT\_16s\_C1IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.28 **NppStatus nppiThreshold\_GT\_16s\_C1R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*)

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.29 **NppStatus nppiThreshold\_GT\_16s\_C3IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.30 **NppStatus nppiThreshold\_GT\_16s\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.31 **NppStatus nppiThreshold\_GT\_16u\_AC4IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.32 NppStatus nppiThreshold\_GT\_16u\_AC4R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3])**

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.33 NppStatus nppiThreshold\_GT\_16u\_C1IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold)**

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.34 **NppStatus nppiThreshold\_GT\_16u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.35 **NppStatus nppiThreshold\_GT\_16u\_C3IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.36 **NppStatus nppiThreshold\_GT\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.37 NppStatus nppiThreshold\_GT\_32f\_AC4IR (Npp32f \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])**

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.38 NppStatus nppiThreshold\_GT\_32f\_AC4R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])**

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.39 **NppStatus nppiThreshold\_GT\_32f\_C1IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.40 **NppStatus nppiThreshold\_GT\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.41 **NppStatus nppiThreshold\_GT\_32f\_C3IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.42 NppStatus nppiThreshold\_GT\_32f\_C3R (const Npp32f \* pSrc, int nSrcStep, Npp32f \* pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])**

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.43 NppStatus nppiThreshold\_GT\_8u\_AC4IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.44 **NppStatus nppiThreshold\_GT\_8u\_AC4R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3])

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.45 **NppStatus nppiThreshold\_GT\_8u\_C1IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.46 **NppStatus nppiThreshold\_GT\_8u\_C1R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.



*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.47 NppStatus nppiThreshold\_GT\_8u\_C3IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.48 NppStatus nppiThreshold\_GT\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.49 **NppStatus nppiThreshold\_GTVal\_16s\_AC4IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3], const Npp16s *rValues*[3])

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.50 **NppStatus nppiThreshold\_GTVal\_16s\_AC4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3], const Npp16s *rValues*[3])

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

##### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.51 **NppStatus nppiThreshold\_GTVal\_16s\_C1IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*, const Npp16s *nValue*)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.52** `NppStatus nppiThreshold_GTVVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.53** `NppStatus nppiThreshold_GTVVal_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.54** `NppStatus nppiThreshold_GTVVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.55** `NppStatus nppiThreshold_GTVVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.56** `NppStatus nppiThreshold_GTVVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.57 **NppStatus nppiThreshold\_GTVa16u\_C1IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, const Npp16u *nValue*)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.58 **NppStatus nppiThreshold\_GTVa16u\_C1R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *nThreshold*, const Npp16u *nValue*)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.59** `NppStatus nppiThreshold_GTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.60** `NppStatus nppiThreshold_GTVal_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.61 **NppStatus nppiThreshold\_GTVAl\_32f\_AC4IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], const Npp32f *rValues*[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.62 **NppStatus nppiThreshold\_GTVAl\_32f\_AC4R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], const Npp32f *rValues*[3])

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.63 **NppStatus nppiThreshold\_GTVAl\_32f\_C1IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*, const Npp32f *nValue*)

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement values.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.64** **NppStatus nppiThreshold\_GTVVal\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*, const Npp32f *nValue*)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.65** **NppStatus nppiThreshold\_GTVVal\_32f\_C3IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3], const Npp32f *rValues*[3])

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.



**7.134.2.66** `NppStatus nppiThreshold_GTVa1_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.67** `NppStatus nppiThreshold_GTVa1_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.68** `NppStatus nppiThreshold_GTVa1_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.69** `NppStatus nppiThreshold_GTVal_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.70** `NppStatus nppiThreshold_GTVal_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.71** `NppStatus nppiThreshold_GTVal_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.72** `NppStatus nppiThreshold_GTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is greater than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

#### 7.134.2.73 **NppStatus nppiThreshold\_LT\_16s\_AC4IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.74 **NppStatus nppiThreshold\_LT\_16s\_AC4R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.75 **NppStatus nppiThreshold\_LT\_16s\_C1IR** (Npp16s \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16s *nThreshold*)

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

##### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.76** `NppStatus nppiThreshold_LT_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.77** `NppStatus nppiThreshold_LT_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.134.2.78 **NppStatus nppiThreshold\_LT\_16s\_C3R** (const Npp16s \* *pSrc*, int *nSrcStep*, Npp16s \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16s *rThresholds*[3])

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.79 **NppStatus nppiThreshold\_LT\_16u\_AC4IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.80 **NppStatus nppiThreshold\_LT\_16u\_AC4R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.81 NppStatus nppiThreshold\_LT\_16u\_C1IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold)**

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.82 NppStatus nppiThreshold\_LT\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold)**

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.83 **NppStatus nppiThreshold\_LT\_16u\_C3IR** (Npp16u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.134.2.84 **NppStatus nppiThreshold\_LT\_16u\_C3R** (const Npp16u \* *pSrc*, int *nSrcStep*, Npp16u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp16u *rThresholds*[3])

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.134.2.85 **NppStatus nppiThreshold\_LT\_32f\_AC4IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).



*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.86** `NppStatus nppiThreshold_LT_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.87** `NppStatus nppiThreshold_LT_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.88 **NppStatus nppiThreshold\_LT\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *nThreshold*)

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.89 **NppStatus nppiThreshold\_LT\_32f\_C3IR** (Npp32f \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.90 **NppStatus nppiThreshold\_LT\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, Npp32f \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp32f *rThresholds*[3])

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.91 **NppStatus nppiThreshold\_LT\_8u\_AC4IR (Npp8u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

#### 7.134.2.92 **NppStatus nppiThreshold\_LT\_8u\_AC4R (const Npp8u \* pSrc, int nSrcStep, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])**

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set value is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.93 **NppStatus nppiThreshold\_LT\_8u\_C1IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.94 **NppStatus nppiThreshold\_LT\_8u\_C1R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.95 **NppStatus nppiThreshold\_LT\_8u\_C3IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3])

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.96** `NppStatus nppiThreshold_LT_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nThreshold, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.97** `NppStatus nppiThreshold_LTVal_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.98** `NppStatus nppiThreshold_LTVal_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.99** `NppStatus nppiThreshold_LTVal_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.100** `NppStatus nppiThreshold_LTVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.101** `NppStatus nppiThreshold_LTVal_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.102** `NppStatus nppiThreshold_LTVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.103** `NppStatus nppiThreshold_LTVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.104** `NppStatus nppiThreshold_LTVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).



### 7.134.2.105 NppStatus nppiThreshold\_LTVal\_16u\_C1IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.106 NppStatus nppiThreshold\_LTVal\_16u\_C1R (const Npp16u \* pSrc, int nSrcStep, Npp16u \* pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue)

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

### 7.134.2.107 NppStatus nppiThreshold\_LTVal\_16u\_C3IR (Npp16u \* pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.108** `NppStatus nppiThreshold_LTVAl_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.109** `NppStatus nppiThreshold_LTVAl_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.110** `NppStatus nppiThreshold_LTVal_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.111** `NppStatus nppiThreshold_LTVal_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.112** `NppStatus nppiThreshold_LTVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThreshold* The threshold value.  
*nValue* The threshold replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.113** `NppStatus nppiThreshold_LTVAl_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.114** `NppStatus nppiThreshold_LTVAl_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.115** `NppStatus nppiThreshold_LTVal_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.116** `NppStatus nppiThreshold_LTVal_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set value is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.134.2.117 **NppStatus nppiThreshold\_LTVal\_8u\_C1IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, const Npp8u *nValue*)

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.134.2.118 **NppStatus nppiThreshold\_LTVal\_8u\_C1R** (const Npp8u \* *pSrc*, int *nSrcStep*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, const Npp8u *nThreshold*, const Npp8u *nValue*)

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThreshold is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

### 7.134.2.119 **NppStatus nppiThreshold\_LTVal\_8u\_C3IR** (Npp8u \* *pSrcDst*, int *nSrcDstStep*, NppiSize *oSizeROI*, const Npp8u *rThresholds*[3], const Npp8u *rValues*[3])

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.120** `NppStatus nppiThreshold_LTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThreshold is true, the pixel is set to rValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholds* The threshold values, one per color channel.  
*rValues* The threshold replacement values, one per color channel.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.121** `NppStatus nppiThreshold_LTValGTVal_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.  
*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.122** `NppStatus nppiThreshold_LTValGTVal_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.123** `NppStatus nppiThreshold_LTValGTVal_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThresholdLT, const Npp16s nValueLT, const Npp16s nThresholdGT, const Npp16s nValueGT)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).



*nThresholdLT* The thresholdLT value.

*nValueLT* The thresholdLT replacement value.

*nThresholdGT* The thresholdGT value.

*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.124** `NppStatus nppiThreshold_LTValGTVal_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThresholdLT, const Npp16s nValueLT, const Npp16s nThresholdGT, const Npp16s nValueGT)`

1 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThresholdLT* The thresholdLT value.

*nValueLT* The thresholdLT replacement value.

*nThresholdGT* The thresholdGT value.

*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.125** `NppStatus nppiThreshold_LTValGTVal_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

3 channel 16-bit signed short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.126** `NppStatus nppiThreshold_LTValGTVal_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholdsLT[3], const Npp16s rValuesLT[3], const Npp16s rThresholdsGT[3], const Npp16s rValuesGT[3])`

3 channel 16-bit signed short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.127** `NppStatus nppiThreshold_LTValGTVal_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.128** `NppStatus nppiThreshold_LTValGTVal_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.129** `NppStatus nppiThreshold_LTValGTVal_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThresholdLT, const Npp16u nValueLT, const Npp16u nThresholdGT, const Npp16u nValueGT)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.130** `NppStatus nppiThreshold_LTValGTVal_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThresholdLT, const Npp16u nValueLT, const Npp16u nThresholdGT, const Npp16u nValueGT)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.131** `NppStatus nppiThreshold_LTValGTVal_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.132** `NppStatus nppiThreshold_LTValGTVal_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholdsLT[3], const Npp16u rValuesLT[3], const Npp16u rThresholdsGT[3], const Npp16u rValuesGT[3])`

3 channel 16-bit unsigned short threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.133** `NppStatus nppiThreshold_LTValGTVal_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.134** `NppStatus nppiThreshold_LTValGTVal_32f_AC4R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.135** `NppStatus nppiThreshold_LTValGTVal_32f_C1IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThresholdLT, const Npp32f nValueLT, const Npp32f nThresholdGT, const Npp32f nValueGT)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.136** `NppStatus nppiThreshold_LTValGTVal_32f_C1R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThresholdLT, const Npp32f nValueLT, const Npp32f nThresholdGT, const Npp32f nValueGT)`

1 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.137** `NppStatus nppiThreshold_LTValGTVal_32f_C3IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

3 channel 32-bit floating point in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.138** `NppStatus nppiThreshold_LTValGTVal_32f_C3R (const Npp32f * pSrc, int nSrcStep, Npp32f * pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholdsLT[3], const Npp32f rValuesLT[3], const Npp32f rThresholdsGT[3], const Npp32f rValuesGT[3])`

3 channel 32-bit floating point threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#).

**7.134.2.139** `NppStatus nppiThreshold_LTValGTVal_8u_AC4IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.



*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.140** `NppStatus nppiThreshold_LTValGTVal_8u_AC4R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set value is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.141** `NppStatus nppiThreshold_LTValGTVal_8u_C1IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThresholdLT, const Npp8u nValueLT, const Npp8u nThresholdGT, const Npp8u nValueGT)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* In-Place-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.142** `NppStatus nppiThreshold_LTValGTVal_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThresholdLT, const Npp8u nValueLT, const Npp8u nThresholdGT, const Npp8u nValueGT)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than nThresholdLT is true, the pixel is set to nValueLT, else if sourcePixel is greater than nThresholdGT the pixel is set to nValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nThresholdLT* The thresholdLT value.  
*nValueLT* The thresholdLT replacement value.  
*nThresholdGT* The thresholdGT value.  
*nValueGT* The thresholdGT replacement value.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.143** `NppStatus nppiThreshold_LTValGTVal_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* Destination-Image Pointer.

*nSrcDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.144** `NppStatus nppiThreshold_LTValGTVal_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholdsLT[3], const Npp8u rValuesLT[3], const Npp8u rThresholdsGT[3], const Npp8u rValuesGT[3])`

3 channel 8-bit unsigned char threshold.

If for a comparison operations sourcePixel is less than rThresholdLT is true, the pixel is set to rValueLT, else if sourcePixel is greater than rThresholdGT the pixel is set to rValueGT, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholdsLT* The thresholdLT values, one per color channel.

*rValuesLT* The thresholdLT replacement values, one per color channel.

*rThresholdsGT* The thresholdGT values, one per channel.

*rValuesGT* The thresholdGT replacement values, one per color channel.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes.

**7.134.2.145** `NppStatus nppiThreshold_Val_16s_AC4IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.146** `NppStatus nppiThreshold_Val_16s_AC4R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.147** `NppStatus nppiThreshold_Val_16s_C1IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.148** `NppStatus nppiThreshold_Val_16s_C1R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s nThreshold, const Npp16s nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.149** `NppStatus nppiThreshold_Val_16s_C3IR (Npp16s * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.150** `NppStatus nppiThreshold_Val_16s_C3R (const Npp16s * pSrc, int nSrcStep, Npp16s * pDst, int nDstStep, NppiSize oSizeROI, const Npp16s rThresholds[3], const Npp16s rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.151** `NppStatus nppiThreshold_Val_16u_AC4IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.152** `NppStatus npptThreshold_Val_16u_AC4R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.153** `NppStatus npptThreshold_Val_16u_C1IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.154** `NppStatus nppiThreshold_Val_16u_C1R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u nThreshold, const Npp16u nValue, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.155** `NppStatus nppiThreshold_Val_16u_C3IR (Npp16u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.



*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.156** `NppStatus nppiThreshold_Val_16u_C3R (const Npp16u * pSrc, int nSrcStep, Npp16u * pDst, int nDstStep, NppiSize oSizeROI, const Npp16u rThresholds[3], const Npp16u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 16-bit unsigned short threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.157** `NppStatus nppiThreshold_Val_32f_AC4IR (Npp32f * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.158** `NppStatus nppiThreshold_Val_32f_AC4R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.159** `NppStatus nppiThreshold_Val_32f_C1IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.160** `NppStatus nppiThreshold_Val_32f_C1R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f nThreshold, const Npp32f nValue, NppCmpOp eComparisonOperation)`

1 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.161** `NppStatus nppiThreshold_Val_32f_C3IR (Npp32f *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.162** `NppStatus nppiThreshold_Val_32f_C3R (const Npp32f *pSrc, int nSrcStep, Npp32f *pDst, int nDstStep, NppiSize oSizeROI, const Npp32f rThresholds[3], const Npp32f rValues[3], NppCmpOp eComparisonOperation)`

3 channel 32-bit floating point threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

Image Data Related Error Codes, ROI Related Error Codes, or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.163** `NppStatus nppiThreshold_Val_8u_AC4IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char in place image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* In-Place Image Pointer.

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.164** `NppStatus nppiThreshold_Val_8u_AC4R (const Npp8u *pSrc, int nSrcStep, Npp8u *pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image threshold, not affecting Alpha.

If for a comparison operations OP the predicate (sourcePixel.channel OP nThreshold) is true, the channel value is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.165** `NppStatus nppiThreshold_Val_8u_C1IR (Npp8u *pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* In-Place-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.166** `NppStatus nppiThreshold_Val_8u_C1R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u nThreshold, const Npp8u nValue, NppCmpOp eComparisonOperation)`

1 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*nThreshold* The threshold value.

*nValue* The threshold replacement value.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.167** `NppStatus nppiThreshold_Val_8u_C3IR (Npp8u * pSrcDst, int nSrcDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char in place threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

**Parameters:**

*pSrcDst* [In-Place Image Pointer](#).

*nSrcDstStep* [In-Place-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

**7.134.2.168** `NppStatus nppiThreshold_Val_8u_C3R (const Npp8u * pSrc, int nSrcStep, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, const Npp8u rThresholds[3], const Npp8u rValues[3], NppCmpOp eComparisonOperation)`

3 channel 8-bit unsigned char threshold.

If for a comparison operations OP the predicate (sourcePixel OP nThreshold) is true, the pixel is set to nValue, otherwise it is set to sourcePixel.

#### Parameters:

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*rThresholds* The threshold values, one per color channel.

*rValues* The threshold replacement values, one per color channel.

*eComparisonOperation* The type of comparison operation to be used. The only valid values are: NPP\_CMP\_LESS and NPP\_CMP\_GREATER.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#), or NPP\_NOT\_SUPPORTED\_MODE\_ERROR if an invalid comparison operation type is specified.

## 7.135 Compare Operations

Compare the pixels of two images and create a binary result image.

### Functions

- **NppStatus nppiCompare\_8u\_C1R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 8-bit unsigned char image compare.*
- **NppStatus nppiCompare\_8u\_C3R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 8-bit unsigned char image compare.*
- **NppStatus nppiCompare\_8u\_C4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char image compare.*
- **NppStatus nppiCompare\_8u\_AC4R** (const **Npp8u** \*pSrc1, int nSrc1Step, const **Npp8u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char image compare, not affecting Alpha.*
- **NppStatus nppiCompare\_16u\_C1R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit unsigned short image compare.*
- **NppStatus nppiCompare\_16u\_C3R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit unsigned short image compare.*
- **NppStatus nppiCompare\_16u\_C4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image compare.*
- **NppStatus nppiCompare\_16u\_AC4R** (const **Npp16u** \*pSrc1, int nSrc1Step, const **Npp16u** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image compare, not affecting Alpha.*
- **NppStatus nppiCompare\_16s\_C1R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit signed short image compare.*
- **NppStatus nppiCompare\_16s\_C3R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit signed short image compare.*
- **NppStatus nppiCompare\_16s\_C4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image compare.*



- **NppStatus nppiCompare\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image compare, not affecting Alpha.*
- **NppStatus nppiCompare\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 32-bit floating point image compare.*
- **NppStatus nppiCompare\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 32-bit floating point image compare.*
- **NppStatus nppiCompare\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit floating point image compare.*
- **NppStatus nppiCompare\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit signed floating point compare, not affecting Alpha.*
- **NppStatus nppiCompareC\_8u\_C1R** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 8-bit unsigned char image compare with constant value.*
- **NppStatus nppiCompareC\_8u\_C3R** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 8-bit unsigned char image compare with constant value.*
- **NppStatus nppiCompareC\_8u\_C4R** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char image compare with constant value.*
- **NppStatus nppiCompareC\_8u\_AC4R** (const **Npp8u** \*pSrc, int nSrcStep, const **Npp8u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 8-bit unsigned char image compare, not affecting Alpha.*
- **NppStatus nppiCompareC\_16u\_C1R** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp16u** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit unsigned short image compare with constant value.*
- **NppStatus nppiCompareC\_16u\_C3R** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp16u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit unsigned short image compare with constant value.*
- **NppStatus nppiCompareC\_16u\_C4R** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp16u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image compare with constant value.*

- **NppStatus** **nppiCompareC\_16u\_AC4R** (const **Npp16u** \*pSrc, int nSrcStep, const **Npp16u** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit unsigned short image compare, not affecting Alpha.*
- **NppStatus** **nppiCompareC\_16s\_C1R** (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 16-bit signed short image compare with constant value.*
- **NppStatus** **nppiCompareC\_16s\_C3R** (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 16-bit signed short image compare with constant value.*
- **NppStatus** **nppiCompareC\_16s\_C4R** (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image compare with constant value.*
- **NppStatus** **nppiCompareC\_16s\_AC4R** (const **Npp16s** \*pSrc, int nSrcStep, const **Npp16s** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 16-bit signed short image compare, not affecting Alpha.*
- **NppStatus** **nppiCompareC\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*1 channel 32-bit floating point image compare with constant value.*
- **NppStatus** **nppiCompareC\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*3 channel 32-bit floating point image compare with constant value.*
- **NppStatus** **nppiCompareC\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit floating point image compare with constant value.*
- **NppStatus** **nppiCompareC\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)  
*4 channel 32-bit signed floating point compare, not affecting Alpha.*
- **NppStatus** **nppiCompareEqualEps\_32f\_C1R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*1 channel 32-bit floating point image compare whether two images are equal within epsilon.*
- **NppStatus** **nppiCompareEqualEps\_32f\_C3R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*3 channel 32-bit floating point image compare whether two images are equal within epsilon.*
- **NppStatus** **nppiCompareEqualEps\_32f\_C4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit floating point image compare whether two images are equal within epsilon.*

- **NppStatus nppiCompareEqualEps\_32f\_AC4R** (const **Npp32f** \*pSrc1, int nSrc1Step, const **Npp32f** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit signed floating point compare whether two images are equal within epsilon, not affecting Alpha.*
- **NppStatus nppiCompareEqualEpsC\_32f\_C1R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** nConstant, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*1 channel 32-bit floating point image compare whether image and constant are equal within epsilon.*
- **NppStatus nppiCompareEqualEpsC\_32f\_C3R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*3 channel 32-bit floating point image compare whether image and constant are equal within epsilon.*
- **NppStatus nppiCompareEqualEpsC\_32f\_C4R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit floating point image compare whether image and constant are equal within epsilon.*
- **NppStatus nppiCompareEqualEpsC\_32f\_AC4R** (const **Npp32f** \*pSrc, int nSrcStep, const **Npp32f** \*pConstants, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **Npp32f** nEpsilon)  
*4 channel 32-bit signed floating point compare whether image and constant are equal within epsilon, not affecting Alpha.*

### 7.135.1 Detailed Description

Compare the pixels of two images and create a binary result image.

In case of multi-channel image types, the condition must be fulfilled for all channels, otherwise the comparison is considered false. The "binary" result image is of type 8u\_C1. False is represented by 0, true by NPP\_MAX\_8U.

### 7.135.2 Function Documentation

**7.135.2.1 NppStatus nppiCompare\_16s\_AC4R** (const **Npp16s** \*pSrc1, int nSrc1Step, const **Npp16s** \*pSrc2, int nSrc2Step, **Npp8u** \*pDst, int nDstStep, **NppiSize** oSizeROI, **NppCmpOp** eComparisonOperation)

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

#### Parameters:

- pSrc1** Source-Image Pointer.
- nSrc1Step** Source-Image Line Step.
- pSrc2** Source-Image Pointer.
- nSrc2Step** Source-Image Line Step.
- pDst** Destination-Image Pointer.
- nDstStep** Destination-Image Line Step.
- oSizeROI** Region-of-Interest (ROI).

***eComparisonOperation*** Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.2** **NppStatus nppiCompare\_16s\_C1R** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

1 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.3** **NppStatus nppiCompare\_16s\_C3R** (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 16-bit signed short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.135.2.4 NppStatus nppiCompare\_16s\_C4R (const Npp16s \* *pSrc1*, int *nSrc1Step*, const Npp16s \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 16-bit signed short image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

#### 7.135.2.5 NppStatus nppiCompare\_16u\_AC4R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

##### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

##### Returns:

Image Data Related Error Codes, ROI Related Error Codes

### 7.135.2.6 NppStatus nppiCompare\_16u\_C1R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

1 channel 16-bit unsigned short image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

### 7.135.2.7 NppStatus nppiCompare\_16u\_C3R (const Npp16u \* *pSrc1*, int *nSrc1Step*, const Npp16u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 16-bit unsigned short image compare.

Compare *pSrc1*'s pixels with corresponding pixels in *pSrc2*.

#### Parameters:

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

#### Returns:

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.8 NppStatus nppiCompare\_16u\_C4R (const Npp16u \* pSrc1, int nSrc1Step, const Npp16u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

4 channel 16-bit unsigned short image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.9 NppStatus nppiCompare\_32f\_AC4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.10 NppStatus nppiCompare\_32f\_C1R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

1 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.11 NppStatus nppiCompare\_32f\_C3R** (const Npp32f \* *pSrc1*, int *nSrc1Step*, const Npp32f \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)



**7.135.2.12 NppStatus nppiCompare\_32f\_C4R (const Npp32f \* pSrc1, int nSrc1Step, const Npp32f \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

4 channel 32-bit floating point image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.13 NppStatus nppiCompare\_8u\_AC4R (const Npp8u \* pSrc1, int nSrc1Step, const Npp8u \* pSrc2, int nSrc2Step, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.14 NppStatus nppiCompare\_8u\_C1R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)**

1 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.15 NppStatus nppiCompare\_8u\_C3R (const Npp8u \* *pSrc1*, int *nSrc1Step*, const Npp8u \* *pSrc2*, int *nSrc2Step*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)**

3 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* [Source-Image Pointer](#).

*nSrc1Step* [Source-Image Line Step](#).

*pSrc2* [Source-Image Pointer](#).

*nSrc2Step* [Source-Image Line Step](#).

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.16** `NppStatus nppiCompare_8u_C4R (const Npp8u * pSrc1, int nSrc1Step, const Npp8u * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare.

Compare pSrc1's pixels with corresponding pixels in pSrc2.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.17** `NppStatus nppiCompareC_16s_AC4R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.18** `NppStatus nppiCompareC_16s_C1R (const Npp16s * pSrc, int nSrcStep, const Npp16s nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*nConstant* constant value.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.19** `NppStatus nppiCompareC_16s_C3R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

3 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.20** `NppStatus nppiCompareC_16s_C4R (const Npp16s * pSrc, int nSrcStep, const Npp16s * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit signed short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.21** `NppStatus nppiCompareC_16u_AC4R (const Npp16u * pSrc, int nSrcStep, const Npp16u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 16-bit unsigned short image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.22** `NppStatus nppiCompareC_16u_C1R (const Npp16u * pSrc, int nSrcStep, const Npp16u nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

1 channel 16-bit unsigned short image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*nConstant* constant value

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.23 NppStatus nppiCompareC\_16u\_C3R (const Npp16u \* *pSrc*, int *nSrcStep*, const Npp16u \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)**

3 channel 16-bit unsigned short image compare with constant value.

Compare *pSrc*'s pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pConstants* pointer to a list of constants, one per color channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.24 NppStatus nppiCompareC\_16u\_C4R (const Npp16u \* *pSrc*, int *nSrcStep*, const Npp16u \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)**

4 channel 16-bit unsigned short image compare with constant value.

Compare *pSrc*'s pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).

*nSrcStep* [Source-Image Line Step](#).

*pConstants* pointer to a list of constants, one per color channel.

*pDst* [Destination-Image Pointer](#).

*nDstStep* [Destination-Image Line Step](#).

*oSizeROI* [Region-of-Interest \(ROI\)](#).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.25 NppStatus nppiCompareC\_32f\_AC4R (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp32f \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)**

4 channel 32-bit signed floating point compare, not affecting Alpha.

Compare *pSrc*'s pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.135.2.26 **NppStatus nppiCompareC\_32f\_C1R** (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp32f *nConstant*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

1 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*nConstant* constant value  
*pDst* [Destination-Image Pointer](#).  
*nDstStep* [Destination-Image Line Step](#).  
*oSizeROI* [Region-of-Interest \(ROI\)](#).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

#### 7.135.2.27 **NppStatus nppiCompareC\_32f\_C3R** (const Npp32f \* *pSrc*, int *nSrcStep*, const Npp32f \* *pConstants*, Npp8u \* *pDst*, int *nDstStep*, NppiSize *oSizeROI*, NppCmpOp *eComparisonOperation*)

3 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* [Source-Image Pointer](#).  
*nSrcStep* [Source-Image Line Step](#).  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* [Destination-Image Pointer](#).

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.28** `NppStatus nppiCompareC_32f_C4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 32-bit floating point image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.29** `NppStatus nppiCompareC_8u_AC4R (const Npp8u * pSrc, int nSrcStep, const Npp8u * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)`

4 channel 8-bit unsigned char image compare, not affecting Alpha.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes



**7.135.2.30 NppStatus nppiCompareC\_8u\_C1R (const Npp8u \* pSrc, int nSrcStep, const Npp8u nConstant, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

1 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*nConstant* constant value.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.31 NppStatus nppiCompareC\_8u\_C3R (const Npp8u \* pSrc, int nSrcStep, const Npp8u \* pConstants, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

3 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constant values, one per color channel..

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.32 NppStatus nppiCompareC\_8u\_C4R (const Npp8u \* pSrc, int nSrcStep, const Npp8u \* pConstants, Npp8u \* pDst, int nDstStep, NppiSize oSizeROI, NppCmpOp eComparisonOperation)**

4 channel 8-bit unsigned char image compare with constant value.

Compare pSrc's pixels with constant value.

**Parameters:**

*pSrc* Source-Image Pointer.  
*nSrcStep* Source-Image Line Step.  
*pConstants* pointer to a list of constants, one per color channel.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*eComparisonOperation* Specifies the comparison operation to be used in the pixel comparison.

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.33** `NppStatus nppiCompareEqualEps_32f_AC4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit signed floating point compare whether two images are equal within epsilon, not affecting Alpha.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.34** `NppStatus nppiCompareEqualEps_32f_C1R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

1 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

*pSrc1* Source-Image Pointer.

*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to pixel absolute differences

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.35** `NppStatus nppiCompareEqualEps_32f_C3R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

3 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.  
*pSrc2* Source-Image Pointer.  
*nSrc2Step* Source-Image Line Step.  
*pDst* Destination-Image Pointer.  
*nDstStep* Destination-Image Line Step.  
*oSizeROI* Region-of-Interest (ROI).  
*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.36** `NppStatus nppiCompareEqualEps_32f_C4R (const Npp32f * pSrc1, int nSrc1Step, const Npp32f * pSrc2, int nSrc2Step, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit floating point image compare whether two images are equal within epsilon.

Compare pSrc1's pixels with corresponding pixels in pSrc2 to determine whether they are equal with a difference of epsilon.

**Parameters:**

*pSrc1* Source-Image Pointer.  
*nSrc1Step* Source-Image Line Step.

*pSrc2* Source-Image Pointer.

*nSrc2Step* Source-Image Line Step.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.37** `NppStatus nppiCompareEqualEpsC_32f_AC4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit signed floating point compare whether image and constant are equal within epsilon, not affecting Alpha.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

**7.135.2.38** `NppStatus nppiCompareEqualEpsC_32f_C1R (const Npp32f * pSrc, int nSrcStep, const Npp32f nConstant, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

1 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*nConstant* constant value

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nEpsilon* epsilon tolerance value to compare to pixel absolute differences

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.39** `NppStatus nppiCompareEqualEpsC_32f_C3R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

3 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

Image Data Related Error Codes, ROI Related Error Codes

**7.135.2.40** `NppStatus nppiCompareEqualEpsC_32f_C4R (const Npp32f * pSrc, int nSrcStep, const Npp32f * pConstants, Npp8u * pDst, int nDstStep, NppiSize oSizeROI, Npp32f nEpsilon)`

4 channel 32-bit floating point image compare whether image and constant are equal within epsilon.

Compare pSrc's pixels with constant value to determine whether they are equal within a difference of epsilon.

**Parameters:**

*pSrc* Source-Image Pointer.

*nSrcStep* Source-Image Line Step.

*pConstants* pointer to a list of constants, one per color channel.

*pDst* Destination-Image Pointer.

*nDstStep* Destination-Image Line Step.

*oSizeROI* Region-of-Interest (ROI).

*nEpsilon* epsilon tolerance value to compare to per color channel pixel absolute differences

**Returns:**

[Image Data Related Error Codes](#), [ROI Related Error Codes](#)

## 7.136 NPP Signal Processing

### Modules

- [Arithmetic and Logical Operations](#)
- [Conversion Functions](#)
- [Filtering Functions](#)

*Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.*

- [Initialization](#)
- [Statistical Functions](#)

*Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.*

- [Memory Management](#)

## 7.137 Arithmetic and Logical Operations

### Modules

- [Arithmetic Operations](#)
- [Logical And Shift Operations](#)



## 7.138 Arithmetic Operations

### Modules

- [AddC](#)  
*Adds a constant value to each sample of a signal.*
- [AddProductC](#)  
*Adds product of a constant and each sample of a source signal to the each sample of destination signal.*
- [MulC](#)  
*Multiplies each sample of a signal by a constant value.*
- [SubC](#)  
*Subtracts a constant from each sample of a signal.*
- [SubCRev](#)  
*Subtracts each sample of a signal from a constant.*
- [DivC](#)  
*Divides each sample of a signal by a constant.*
- [DivCRev](#)  
*Divides a constant by each sample of a signal.*
- [Add](#)  
*Sample by sample addition of two signals.*
- [AddProduct](#)  
*Adds sample by sample product of two signals to the destination signal.*
- [Mul](#)  
*Sample by sample multiplication the samples of two signals.*
- [Sub](#)  
*Sample by sample subtraction of the samples of two signals.*
- [Div](#)  
*Sample by sample division of the samples of two signals.*
- [Div\\_Round](#)  
*Sample by sample division of the samples of two signals with rounding.*
- [Abs](#)  
*Absolute value of each sample of a signal.*
- [Sqr](#)  
*Squares each sample of a signal.*
- [Sqrt](#)

*Square root of each sample of a signal.*

- [Cubrt](#)

*Cube root of each sample of a signal.*

- [Exp](#)

*E raised to the power of each sample of a signal.*

- [Ln](#)

*Natural logarithm of each sample of a signal.*

- [10Log10](#)

*Ten times the decimal logarithm of each sample of a signal.*

- [SumLn](#)

*Sums up the natural logarithm of each sample of a signal.*

- [Arctan](#)

*Inverse tangent of each sample of a signal.*

- [Normalize](#)

*Normalize each sample of a real or complex signal using offset and division operations.*

- [Cauchy, CauchyD, and CauchyDD2](#)

*Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.*

## 7.139 AddC

Adds a constant value to each sample of a signal.

### Functions

- **NppStatus nppsAddC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal add constant, scale, then clamp to saturated value*
- **NppStatus nppsAddC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned charvector add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short vector add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.*
- **NppStatus nppsAddC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal add constant and scale.*
- **NppStatus nppsAddC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integersignal add constant and scale.*
- **NppStatus nppsAddC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.*
- **NppStatus nppsAddC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

- [NppStatus nppsAddC\\_32f\\_I](#) ([Npp32f](#) nValue, [Npp32f](#) \*pSrcDst, int nLength)  
32-bit floating point in place signal add constant.
- [NppStatus nppsAddC\\_32f](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) nValue, [Npp32f](#) \*pDst, int nLength)  
32-bit floating point signal add constant.
- [NppStatus nppsAddC\\_32fc\\_I](#) ([Npp32fc](#) nValue, [Npp32fc](#) \*pSrcDst, int nLength)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.
- [NppStatus nppsAddC\\_32fc](#) (const [Npp32fc](#) \*pSrc, [Npp32fc](#) nValue, [Npp32fc](#) \*pDst, int nLength)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.
- [NppStatus nppsAddC\\_64f\\_I](#) ([Npp64f](#) nValue, [Npp64f](#) \*pSrcDst, int nLength)  
64-bit floating point, in place signal add constant.
- [NppStatus nppsAddC\\_64f](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) nValue, [Npp64f](#) \*pDst, int nLength)  
64-bit floating pointsignal add constant.
- [NppStatus nppsAddC\\_64fc\\_I](#) ([Npp64fc](#) nValue, [Npp64fc](#) \*pSrcDst, int nLength)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.
- [NppStatus nppsAddC\\_64fc](#) (const [Npp64fc](#) \*pSrc, [Npp64fc](#) nValue, [Npp64fc](#) \*pDst, int nLength)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

### 7.139.1 Detailed Description

Adds a constant value to each sample of a signal.

### 7.139.2 Function Documentation

#### 7.139.2.1 [NppStatus nppsAddC\\_16s\\_ISfs](#) ([Npp16s](#) nValue, [Npp16s](#) \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal add constant, scale, then clamp to saturated value.

##### Parameters:

[pSrcDst](#) [In-Place Signal Pointer](#).

[nValue](#) Constant value to be added to each vector element

[nLength](#) [Signal Length](#).

[nScaleFactor](#) [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.2 NppStatus nppsAddC\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.3 NppStatus nppsAddC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.4 NppStatus nppsAddC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be added to each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.5 NppStatus nppsAddC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.6 NppStatus nppsAddC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short vector add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.7 NppStatus nppsAddC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal add constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.8 NppStatus nppsAddC\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal add constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.9 NppStatus nppsAddC\_32fc (const Npp32fc \* pSrc, Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal add constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.10 NppStatus nppsAddC\_32fc\_I (Npp32fc nValue, Npp32fc \* pSrcDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal add constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.11 NppStatus nppsAddC\_32s\_ISfs (Npp32s nValue, Npp32s \* pSrcDst, int nLength, int nScaleFactor)**

32-bit signed integer in place signal add constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be added to each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.139.2.12 NppStatus nppsAddC\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integersignal add constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be added to each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.139.2.13 NppStatus nppsAddC\_32sc\_ISfs (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal add constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be added to each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



**7.139.2.14 NppStatus nppsAddC\_32sc\_Sfs (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) signal add constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.139.2.15 NppStatus nppsAddC\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating pointsignal add constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be added to each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.139.2.16 NppStatus nppsAddC\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point, in place signal add constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be added to each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.139.2.17 NppStatus nppsAddC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal add constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.18 NppStatus nppsAddC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal add constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.19 NppStatus nppsAddC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal add constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.139.2.20** **NppStatus nppsAddC\_8u\_Sfs** (**const Npp8u \* pSrc**, **Npp8u nValue**, **Npp8u \* pDst**, **int nLength**, **int nScaleFactor**)

8-bit unsigned charvector add constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added to each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.140 AddProductC

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

### Functions

- **NppStatus** **nppsAddProductC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)

*32-bit floating point signal add product of signal times constant to destination signal.*

### 7.140.1 Detailed Description

Adds product of a constant and each sample of a source signal to the each sample of destination signal.

### 7.140.2 Function Documentation

**7.140.2.1** **NppStatus nppsAddProductC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)

32-bit floating point signal add product of signal times constant to destination signal.

#### Parameters:

*pSrc* **Source Signal Pointer.**

*nValue* Constant value to be multiplied by each vector element

*pDst* **Destination Signal Pointer.**

*nLength* **Signal Length.**

#### Returns:

**Signal Data Related Error Codes, Length Related Error Codes.**

## 7.141 MulC

Multiplies each sample of a signal by a constant value.

### Functions

- **NppStatus nppsMulC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal times constant, scale, then clamp to saturated value*
- **NppStatus nppsMulC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary)signal times constant, scale, then clamp to saturated value.*
- **NppStatus nppsMulC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal times constant and scale.*
- **NppStatus nppsMulC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal times constant and scale.*
- **NppStatus nppsMulC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.*
- **NppStatus nppsMulC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.

- **NppStatus nppsMulC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
32-bit floating point in place signal times constant.
- **NppStatus nppsMulC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
32-bit floating point signal times constant.
- **NppStatus nppsMulC\_Low\_32f16s** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp16s** \*pDst, int nLength)  
32-bit floating point signal times constant with output converted to 16-bit signed integer.
- **NppStatus nppsMulC\_32f16s\_Sfs** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.
- **NppStatus nppsMulC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.
- **NppStatus nppsMulC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.
- **NppStatus nppsMulC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
64-bit floating point, in place signal times constant.
- **NppStatus nppsMulC\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
64-bit floating point signal times constant.
- **NppStatus nppsMulC\_64f64s\_ISfs** (**Npp64f** nValue, **Npp64s** \*pDst, int nLength, int nScaleFactor)  
64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.
- **NppStatus nppsMulC\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.
- **NppStatus nppsMulC\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.

### 7.141.1 Detailed Description

Multiplies each sample of a signal by a constant value.

### 7.141.2 Function Documentation

#### 7.141.2.1 **NppStatus nppsMulC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be multiplied by each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.141.2.2 NppStatus nppsMulC\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be multiplied by each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.141.2.3 NppStatus nppsMulC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be multiplied by each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.141.2.4 NppStatus nppsMulC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.5 NppStatus nppsMulC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.6 NppStatus nppsMulC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.141.2.7 NppStatus nppsMulC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal times constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.8 NppStatus nppsMulC\_32f16s\_Sfs (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit floating point signal times constant with output converted to 16-bit signed integer with scaling and saturation of output result.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nScaleFactor* [Integer Result Scaling](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.9 NppStatus nppsMulC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal times constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.10 NppStatus nppsMulC\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal times constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.11 NppStatus nppsMulC\_32fc\_I (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal times constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.12 NppStatus nppsMulC\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal times constant and scale.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.13 NppStatus nppsMulC\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal times constant and scale.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.14 NppStatus nppsMulC\_32sc\_ISfs (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal times constant and scale.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.15 NppStatus nppsMulC\_32sc\_Sfs (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) signal times constant and scale.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.16 NppStatus nppsMulC\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal times constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.17 NppStatus nppsMulC\_64f64s\_ISfs (Npp64f *nValue*, Npp64s \* *pDst*, int *nLength*, int *nScaleFactor*)**

64-bit floating point signal times constant with in place conversion to 64-bit signed integer and with scaling and saturation of output result.

**Parameters:**

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.18 NppStatus nppsMulC\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point, in place signal times constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.19 NppStatus nppsMulC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal times constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.20 NppStatus nppsMulC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal times constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.21 NppStatus nppsMulC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal times constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be multiplied by each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.141.2.22 NppStatus nppsMulC\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal times constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.141.2.23 NppStatus nppsMulC\_Low\_32f16s (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp16s \* *pDst*, int *nLength*)**

32-bit floating point signal times constant with output converted to 16-bit signed integer.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be multiplied by each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.142 SubC

Subtracts a constant from each sample of a signal.

### Functions

- **NppStatus nppsSubC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value*
- **NppStatus nppsSubC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.*
- **NppStatus nppsSubC\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal subtract constant and scale.*
- **NppStatus nppsSubC\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal subtract constant and scale.*
- **NppStatus nppsSubC\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.*
- **NppStatus nppsSubC\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

- **NppStatus nppsSubC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
32-bit floating point in place signal subtract constant.
- **NppStatus nppsSubC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
32-bit floating point signal subtract constant.
- **NppStatus nppsSubC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.
- **NppStatus nppsSubC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.
- **NppStatus nppsSubC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)  
64-bit floating point, in place signal subtract constant.
- **NppStatus nppsSubC\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
64-bit floating point signal subtract constant.
- **NppStatus nppsSubC\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.
- **NppStatus nppsSubC\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

### 7.142.1 Detailed Description

Subtracts a constant from each sample of a signal.

### 7.142.2 Function Documentation

#### 7.142.2.1 **NppStatus nppsSubC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal subtract constant, scale, then clamp to saturated value.

##### Parameters:

**pSrcDst** [In-Place Signal Pointer](#).

**nValue** Constant value to be subtracted from each vector element

**nLength** [Signal Length](#).

**nScaleFactor** [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



### 7.142.2.2 NppStatus nppsSubC\_16s\_Sfs (const Npp16s \* *pSrc*, Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.142.2.3 NppStatus nppsSubC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.142.2.4 NppStatus nppsSubC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract constant, scale, then clamp to saturated value.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.5 NppStatus nppsSubC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.142.2.6 NppStatus nppsSubC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.142.2.7 NppStatus nppsSubC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal subtract constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be subtracted from each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.142.2.8 NppStatus nppsSubC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal subtract constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.9 NppStatus nppsSubC\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.10 NppStatus nppsSubC\_32fc\_I (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.11 NppStatus nppsSubC\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal subtract constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.142.2.12** `NppStatus nppsSubC_32s_Sfs (const Npp32s * pSrc, Npp32s nValue, Npp32s * pDst, int nLength, int nScaleFactor)`

32-bit signed integer signal subtract constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.142.2.13** `NppStatus nppsSubC_32sc_ISfs (Npp32sc nValue, Npp32sc * pSrcDst, int nLength, int nScaleFactor)`

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be subtracted from each vector element  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.142.2.14 **NppStatus nppsSubC\_32sc\_Sfs** (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract constant and scale.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.142.2.15 **NppStatus nppsSubC\_64f** (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal subtract constant.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.142.2.16 **NppStatus nppsSubC\_64f\_I** (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)

64-bit floating point, in place signal subtract constant.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* Length of the vectors, number of items.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.17 NppStatus nppsSubC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.18 NppStatus nppsSubC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.19 NppStatus nppsSubC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal subtract constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.142.2.20 NppStatus nppsSubC\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal subtract constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be subtracted from each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.143 SubCRev

Subtracts each sample of a signal from a constant.

### Functions

- **NppStatus** **nppsSubCRev\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value*
- **NppStatus** **nppsSubCRev\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus** **nppsSubCRev\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus** **nppsSubCRev\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus** **nppsSubCRev\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus** **nppsSubCRev\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus** **nppsSubCRev\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus** **nppsSubCRev\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.*
- **NppStatus** **nppsSubCRev\_32s\_ISfs** (**Npp32s** nValue, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal subtract from constant and scale.*
- **NppStatus** **nppsSubCRev\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** nValue, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integersignal subtract from constant and scale.*
- **NppStatus** **nppsSubCRev\_32sc\_ISfs** (**Npp32sc** nValue, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)



32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.

- **NppStatus nppsSubCRev\_32sc\_Sfs** (const **Npp32sc** \*pSrc, **Npp32sc** nValue, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.

- **NppStatus nppsSubCRev\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)

32-bit floating point in place signal subtract from constant.

- **NppStatus nppsSubCRev\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)

32-bit floating point signal subtract from constant.

- **NppStatus nppsSubCRev\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.

- **NppStatus nppsSubCRev\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.

- **NppStatus nppsSubCRev\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)

64-bit floating point, in place signal subtract from constant.

- **NppStatus nppsSubCRev\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)

64-bit floating point signal subtract from constant.

- **NppStatus nppsSubCRev\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.

- **NppStatus nppsSubCRev\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.

### 7.143.1 Detailed Description

Subtracts each sample of a signal from a constant.

### 7.143.2 Function Documentation

#### 7.143.2.1 **NppStatus nppsSubCRev\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal subtract from constant, scale, then clamp to saturated value.

#### Parameters:

**pSrcDst** In-Place Signal Pointer.

**nValue** Constant value each vector element is to be subtracted from

**nLength** Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.2** `NppStatus nppsSubCRev_16s_Sfs (const Npp16s * pSrc, Npp16s nValue, Npp16s * pDst, int nLength, int nScaleFactor)`

16-bit signed short signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.3** `NppStatus nppsSubCRev_16sc_ISfs (Npp16sc nValue, Npp16sc * pSrcDst, int nLength, int nScaleFactor)`

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.4** `NppStatus nppsSubCRev_16sc_Sfs (const Npp16sc * pSrc, Npp16sc nValue, Npp16sc * pDst, int nLength, int nScaleFactor)`

16-bit integer complex number (16 bit real, 16 bit imaginary) signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.5 NppStatus nppsSubCRev\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.6 NppStatus nppsSubCRev\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.7 NppStatus nppsSubCRev\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.8 NppStatus nppsSubCRev\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.9 NppStatus nppsSubCRev\_32fc (const Npp32fc \* pSrc, Npp32fc nValue, Npp32fc \* pDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.10 NppStatus nppsSubCRev\_32fc\_I (Npp32fc nValue, Npp32fc \* pSrcDst, int nLength)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value each vector element is to be subtracted from  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.11 NppStatus nppsSubCRev\_32s\_ISfs (Npp32s *nValue*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal subtract from constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.12 NppStatus nppsSubCRev\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integersignal subtract from constant and scale.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.13 NppStatus nppsSubCRev\_32sc\_ISfs (Npp32sc *nValue*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit integer complex number (32 bit real, 32 bit imaginary) in place signal subtract from constant and scale.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.143.2.14 **NppStatus nppsSubCRev\_32sc\_Sfs** (const Npp32sc \* *pSrc*, Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit integer complex number (32 bit real, 32 bit imaginary) signal subtract from constant and scale.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.143.2.15 **NppStatus nppsSubCRev\_64f** (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal subtract from constant.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.143.2.16 **NppStatus nppsSubCRev\_64f\_I** (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)

64-bit floating point, in place signal subtract from constant.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*nLength* Length of the vectors, number of items.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.143.2.17 NppStatus nppsSubCRev\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal subtract from constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.18 NppStatus nppsSubCRev\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal subtract from constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.19 NppStatus nppsSubCRev\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal subtract from constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value each vector element is to be subtracted from

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.143.2.20 NppStatus nppsSubCRev\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal subtract from constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value each vector element is to be subtracted from

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



## 7.144 DivC

Divides each sample of a signal by a constant.

### Functions

- **NppStatus nppsDivC\_8u\_ISfs** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value*
- **NppStatus nppsDivC\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16u\_ISfs** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16sc\_ISfs** (**Npp16sc** nValue, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** nValue, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.*
- **NppStatus nppsDivC\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place signal divided by constant.*
- **NppStatus nppsDivC\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal divided by constant.*
- **NppStatus nppsDivC\_32fc\_I** (**Npp32fc** nValue, **Npp32fc** \*pSrcDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.*
- **NppStatus nppsDivC\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** nValue, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.*
- **NppStatus nppsDivC\_64f\_I** (**Npp64f** nValue, **Npp64f** \*pSrcDst, int nLength)

*64-bit floating point in place signal divided by constant.*

- **NppStatus nppsDivC\_64f** (const **Npp64f** \*pSrc, **Npp64f** nValue, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal divided by constant.*
- **NppStatus nppsDivC\_64fc\_I** (**Npp64fc** nValue, **Npp64fc** \*pSrcDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.*
- **NppStatus nppsDivC\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** nValue, **Npp64fc** \*pDst, int nLength)  
*64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.*

### 7.144.1 Detailed Description

Divides each sample of a signal by a constant.

### 7.144.2 Function Documentation

#### 7.144.2.1 **NppStatus nppsDivC\_16s\_ISfs** (**Npp16s** nValue, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal divided by constant, scale, then clamp to saturated value.

##### Parameters:

**pSrcDst** [In-Place Signal Pointer](#).

**nValue** Constant value to be divided into each vector element

**nLength** [Signal Length](#).

**nScaleFactor** [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.144.2.2 **NppStatus nppsDivC\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** nValue, **Npp16s** \*pDst, int nLength, int nScaleFactor)

16-bit signed short signal divided by constant, scale, then clamp to saturated value.

##### Parameters:

**pSrc** [Source Signal Pointer](#).

**nValue** Constant value to be divided into each vector element

**pDst** [Destination Signal Pointer](#).

**nLength** [Signal Length](#).

**nScaleFactor** [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.3 NppStatus nppsDivC\_16sc\_ISfs (Npp16sc *nValue*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.4 NppStatus nppsDivC\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit integer complex number (16 bit real, 16 bit imaginary) signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.5 NppStatus nppsDivC\_16u\_ISfs (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.6 NppStatus nppsDivC\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.7 NppStatus nppsDivC\_32f (const Npp32f \* *pSrc*, Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal divided by constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.8 NppStatus nppsDivC\_32f\_I (Npp32f *nValue*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.9 NppStatus nppsDivC\_32fc (const Npp32fc \* *pSrc*, Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.144.2.10 NppStatus nppsDivC\_32fc\_I (Npp32fc *nValue*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit floating point complex number (32 bit real, 32 bit imaginary) in place signal divided by constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be divided into each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.144.2.11 NppStatus nppsDivC\_64f (const Npp64f \* *pSrc*, Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal divided by constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be divided into each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.144.2.12 NppStatus nppsDivC\_64f\_I (Npp64f *nValue*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* Length of the vectors, number of items.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.13 NppStatus nppsDivC\_64fc (const Npp64fc \* *pSrc*, Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) signal divided by constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.14 NppStatus nppsDivC\_64fc\_I (Npp64fc *nValue*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit floating point complex number (64 bit real, 64 bit imaginary) in place signal divided by constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.15 NppStatus nppsDivC\_8u\_ISfs (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal divided by constant, scale, then clamp to saturated value

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.144.2.16** `NppStatus nppsDivC_8u_Sfs (const Npp8u * pSrc, Npp8u nValue, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal divided by constant, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be divided into each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.145 DivCRev

Divides a constant by each sample of a signal.

### Functions

- **NppStatus nppsDivCRev\_16u\_I** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place constant divided by signal, then clamp to saturated value.*
- **NppStatus nppsDivCRev\_16u** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal divided by constant, then clamp to saturated value.*
- **NppStatus nppsDivCRev\_32f\_I** (**Npp32f** nValue, **Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point in place constant divided by signal.*
- **NppStatus nppsDivCRev\_32f** (const **Npp32f** \*pSrc, **Npp32f** nValue, **Npp32f** \*pDst, int nLength)  
*32-bit floating point constant divided by signal.*

### 7.145.1 Detailed Description

Divides a constant by each sample of a signal.

### 7.145.2 Function Documentation

#### 7.145.2.1 NppStatus nppsDivCRev\_16u (const Npp16u \* pSrc, Npp16u nValue, Npp16u \* pDst, int nLength)

16-bit unsigned short signal divided by constant, then clamp to saturated value.

##### Parameters:

- pSrc* **Source Signal Pointer.**  
*nValue* **Constant value to be divided by each vector element**  
*pDst* **Destination Signal Pointer.**  
*nLength* **Signal Length.**

##### Returns:

**Signal Data Related Error Codes, Length Related Error Codes.**

#### 7.145.2.2 NppStatus nppsDivCRev\_16u\_I (Npp16u nValue, Npp16u \* pSrcDst, int nLength)

16-bit unsigned short in place constant divided by signal, then clamp to saturated value.

##### Parameters:

- pSrcDst* **In-Place Signal Pointer.**



*nValue* Constant value to be divided by each vector element  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.2.3 NppStatus nppsDivCRev\_32f (const Npp32f \* pSrc, Npp32f nValue, Npp32f \* pDst, int nLength)**

32-bit floating point constant divided by signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*nValue* Constant value to be divided by each vector element  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.145.2.4 NppStatus nppsDivCRev\_32f\_I (Npp32f nValue, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place constant divided by signal.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).  
*nValue* Constant value to be divided by each vector element  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.146 Add

Sample by sample addition of two signals.

### Functions

- **NppStatus nppsAdd\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal add signal, then clamp to saturated value.*
- **NppStatus nppsAdd\_16u** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal add signal, then clamp to saturated value.*
- **NppStatus nppsAdd\_32u** (const **Npp32u** \*pSrc1, const **Npp32u** \*pSrc2, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned int signal add signal, then clamp to saturated value.*
- **NppStatus nppsAdd\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal add signal, then clamp to saturated value.*
- **NppStatus nppsAdd\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal add signal, then clamp to saturated value.*
- **NppStatus nppsAdd\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal add signal, then clamp to saturated value.*
- **NppStatus nppsAdd\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal add signal, then clamp to saturated value.*
- **NppStatus nppsAdd\_8u16u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.*
- **NppStatus nppsAdd\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.*
- **NppStatus nppsAdd\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char add signal, scale, then clamp to saturated value.*
- **NppStatus nppsAdd\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short add signal, scale, then clamp to saturated value.*
- **NppStatus nppsAdd\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_32s_Sfs` (const `Npp32s` \*pSrc1, const `Npp32s` \*pSrc2, `Npp32s` \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_64s_Sfs` (const `Npp64s` \*pSrc1, const `Npp64s` \*pSrc2, `Npp64s` \*pDst, int nLength, int nScaleFactor)

*64-bit signed integer add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_16sc_Sfs` (const `Npp16sc` \*pSrc1, const `Npp16sc` \*pSrc2, `Npp16sc` \*pDst, int nLength, int nScaleFactor)

*16-bit signed complex short add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_32sc_Sfs` (const `Npp32sc` \*pSrc1, const `Npp32sc` \*pSrc2, `Npp32sc` \*pDst, int nLength, int nScaleFactor)

*32-bit signed complex integer add signal, scale, then clamp to saturated value.*

- `NppStatus nppsAdd_16s_I` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength)

*16-bit signed short in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_32f_I` (const `Npp32f` \*pSrc, `Npp32f` \*pSrcDst, int nLength)

*32-bit floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_64f_I` (const `Npp64f` \*pSrc, `Npp64f` \*pSrcDst, int nLength)

*64-bit floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_32fc_I` (const `Npp32fc` \*pSrc, `Npp32fc` \*pSrcDst, int nLength)

*32-bit complex floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_64fc_I` (const `Npp64fc` \*pSrc, `Npp64fc` \*pSrcDst, int nLength)

*64-bit complex floating point in place signal add signal, then clamp to saturated value.*

- `NppStatus nppsAdd_16s32s_I` (const `Npp16s` \*pSrc, `Npp32s` \*pSrcDst, int nLength)

*16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.*

- `NppStatus nppsAdd_8u_ISfs` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength, int nScaleFactor)

*8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsAdd_16u_ISfs` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength, int nScaleFactor)

*16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsAdd_16s_ISfs` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength, int nScaleFactor)

*16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.*

- `NppStatus nppsAdd_32s_ISfs` (const `Npp32s` \*pSrc, `Npp32s` \*pSrcDst, int nLength, int nScaleFactor)

*32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsAdd\_16sc\_ISfs** (const [Npp16sc](#) \*pSrc, [Npp16sc](#) \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsAdd\_32sc\_ISfs** (const [Npp32sc](#) \*pSrc, [Npp32sc](#) \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.*

### 7.146.1 Detailed Description

Sample by sample addition of two signals.

### 7.146.2 Function Documentation

#### 7.146.2.1 **NppStatus nppsAdd\_16s** (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, [Npp16s](#) \*pDst, int nLength)

16-bit signed short signal add signal, then clamp to saturated value.

##### Parameters:

[pSrc1](#) [Source Signal Pointer](#).  
[pSrc2](#) [Source Signal Pointer](#). signal2 elements to be added to signal1 elements  
[pDst](#) [Destination Signal Pointer](#).  
[nLength](#) [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.146.2.2 **NppStatus nppsAdd\_16s32f** (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, [Npp32f](#) \*pDst, int nLength)

16-bit signed short signal add signal with 32-bit floating point result, then clamp to saturated value.

##### Parameters:

[pSrc1](#) [Source Signal Pointer](#).  
[pSrc2](#) [Source Signal Pointer](#). signal2 elements to be added to signal1 elements  
[pDst](#) [Destination Signal Pointer](#).  
[nLength](#) [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.2.3 NppStatus nppsAdd\_16s32s\_I (const Npp16s \* pSrc, Npp32s \* pSrcDst, int nLength)**

16/32-bit signed short in place signal add signal with 32-bit signed integer results, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.4 NppStatus nppsAdd\_16s\_I (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength)**

16-bit signed short in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.5 NppStatus nppsAdd\_16s\_ISfs (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength, int nScaleFactor)**

16-bit signed short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.6 NppStatus nppsAdd\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.7 NppStatus nppsAdd\_16sc\_ISfs (const Npp16sc \* *pSrc*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.8 NppStatus nppsAdd\_16sc\_Sfs (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed complex short add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.9 NppStatus nppsAdd\_16u (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.10 NppStatus nppsAdd\_16u\_ISfs (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.11 NppStatus nppsAdd\_16u\_Sfs (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.12 NppStatus nppsAdd\_32f (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.13 NppStatus nppsAdd\_32f\_I (const Npp32f \* *pSrc*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.14 NppStatus nppsAdd\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



**7.146.2.15 NppStatus nppsAdd\_32fc\_I (const Npp32fc \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.2.16 NppStatus nppsAdd\_32s\_ISfs (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.2.17 NppStatus nppsAdd\_32s\_Sfs (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.146.2.18 NppStatus nppsAdd\_32sc\_ISfs (const Npp32sc \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit complex signed integer in place signal add signal, with scaling, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.146.2.19 NppStatus nppsAdd\_32sc\_Sfs (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed complex integer add signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.146.2.20 NppStatus nppsAdd\_32u (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, Npp32u \* *pDst*, int *nLength*)

32-bit unsigned int signal add signal, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.21 NppStatus nppsAdd\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.22 NppStatus nppsAdd\_64f\_I (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be added to signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.23 NppStatus nppsAdd\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be added to signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.146.2.24 NppStatus nppsAdd\_64fc\_I (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point in place signal add signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.2.25 NppStatus nppsAdd\_64s\_Sfs (const Npp64s \* *pSrc1*, const Npp64s \* *pSrc2*, Npp64s \* *pDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.2.26 NppStatus nppsAdd\_8u16u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)**

8-bit unsigned char signal add signal with 16-bit unsigned result, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be added to signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.2.27 NppStatus nppsAdd\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal add signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be added to signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.146.2.28 NppStatus nppsAdd\_8u\_Sfs (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char add signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be added to signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.147 AddProduct

Adds sample by sample product of two signals to the destination signal.

### Functions

- **NppStatus** **nppsAddProduct\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)

*32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)

*64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)

*32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)

*64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.*

- **NppStatus** **nppsAddProduct\_16s32s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.*

### 7.147.1 Detailed Description

Adds sample by sample product of two signals to the destination signal.

## 7.147.2 Function Documentation

### 7.147.2.1 NppStatus nppsAddProduct\_16s32s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal add product of source signal1 times source signal2 to 32-bit signed integer destination signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.147.2.2 NppStatus nppsAddProduct\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* Destination Signal Pointer. product of source1 and source2 signal elements to be added to destination elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.147.2.3 NppStatus nppsAddProduct\_32f (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

32-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.147.2.4 NppStatus nppsAddProduct\_32fc (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.147.2.5 NppStatus nppsAddProduct\_32s\_Sfs (const Npp32s \* pSrc1, const Npp32s \* pSrc2, Npp32s \* pDst, int nLength, int nScaleFactor)**

32-bit signed short signal add product of source signal1 times source signal2 to destination signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.147.2.6 NppStatus nppsAddProduct\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.147.2.7 NppStatus nppsAddProduct\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal add product of source signal times destination signal to destination signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#). product of source1 and source2 signal elements to be added to destination elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.148 Mul

Sample by sample multiplication the samples of two signals.

### Functions

- **NppStatus nppsMul\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal times signal, then clamp to saturated value.*
- **NppStatus nppsMul\_8u16u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.*
- **NppStatus nppsMul\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.*
- **NppStatus nppsMul\_32f32fc** (const **Npp32f** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.*
- **NppStatus nppsMul\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal time signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal times signal, scale, then clamp to saturated value.*
- **NppStatus nppsMul\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)

*16-bit signed complex short signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_32sc\_Sfs** (const **Npp32sc** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

*32-bit signed complex integer signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_16u16s\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)

*16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.*

- **NppStatus nppsMul\_16s32s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.*

- **NppStatus nppsMul\_32s32sc\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.*

- **NppStatus nppsMul\_Low\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)

*32-bit signed integer signal times signal, scale, then clamp to saturated value.*

- **NppStatus nppsMul\_16s\_I** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength)

*16-bit signed short in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)

*32-bit floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)

*64-bit floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)

*32-bit complex floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)

*64-bit complex floating point in place signal times signal, then clamp to saturated value.*

- **NppStatus nppsMul\_32f32fc\_I** (const **Npp32f** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)

*32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.*

- **NppStatus nppsMul\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)

*8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsMul\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32sc\_ISfs** (const **Npp32sc** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsMul\_32s32sc\_ISfs** (const **Npp32s** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.*

### 7.148.1 Detailed Description

Sample by sample multiplication the samples of two signals.

### 7.148.2 Function Documentation

#### 7.148.2.1 **NppStatus nppsMul\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)

16-bit signed short signal times signal, then clamp to saturated value.

##### Parameters:

**pSrc1** [Source Signal Pointer](#).

**pSrc2** [Source Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

**pDst** [Destination Signal Pointer](#).

**nLength** [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.148.2.2 NppStatus nppsMul\_16s32f (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

16-bit signed short signal times signal with 32-bit floating point result, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.148.2.3 NppStatus nppsMul\_16s32s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal times signal, scale, then clamp to 32-bit signed saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.148.2.4 NppStatus nppsMul\_16s\_I (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*)

16-bit signed short in place signal times signal, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.5 NppStatus nppsMul\_16s\_ISfs (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.2.6 NppStatus nppsMul\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit signed short signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.2.7 NppStatus nppsMul\_16sc\_ISfs (const Npp16sc \* *pSrc*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.148.2.8 NppStatus nppsMul\_16sc\_Sfs (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short signal times signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.148.2.9 NppStatus nppsMul\_16u16s\_Sfs (const Npp16u \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal times 16-bit signed short signal, scale, then clamp to 16-bit signed saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.148.2.10 NppStatus nppsMul\_16u\_ISfs (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short in place signal times signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.148.2.11 **NppStatus nppsMul\_16u\_Sfs** (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit unsigned short signal time signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.148.2.12 **NppStatus nppsMul\_32f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

32-bit floating point signal times signal, then clamp to saturated value.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.148.2.13 **NppStatus nppsMul\_32f32fc** (const Npp32f \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)

32-bit floating point signal times 32-bit complex floating point signal with complex 32-bit floating point result, then clamp to saturated value.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.148.2.14 NppStatus nppsMul\_32f32fc\_I (const Npp32f \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal times 32-bit floating point signal, then clamp to 32-bit complex floating point saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.15 NppStatus nppsMul\_32f\_I (const Npp32f \* *pSrc*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.16 NppStatus nppsMul\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.17 NppStatus nppsMul\_32fc\_I (const Npp32fc \* *pSrc*, Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.2.18 NppStatus nppsMul\_32s32sc\_ISfs (const Npp32s \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit complex signed integer in place signal times 32-bit signed integer signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.2.19 NppStatus nppsMul\_32s32sc\_Sfs (const Npp32s \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal times 32-bit complex signed integer signal, scale, then clamp to 32-bit complex integer saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.148.2.20 NppStatus nppsMul\_32s\_ISfs (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.21 NppStatus nppsMul\_32s\_Sfs (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal times signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.22 NppStatus nppsMul\_32sc\_ISfs (const Npp32sc \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit complex signed integer in place signal times signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.148.2.23 **NppStatus nppsMul\_32sc\_Sfs** (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed complex integer signal times signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.148.2.24 **NppStatus nppsMul\_64f** (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal times signal, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.148.2.25 **NppStatus nppsMul\_64f\_I** (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)

64-bit floating point in place signal times signal, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer, signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.26 NppStatus nppsMul\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.27 NppStatus nppsMul\_64fc\_I (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point in place signal times signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be multiplied by signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.148.2.28 NppStatus nppsMul\_8u16u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*)**

8-bit unsigned char signal times signal with 16-bit unsigned result, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be multiplied by signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.148.2.29 **NppStatus nppsMul\_8u\_ISfs** (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char in place signal times signal, with scaling, then clamp to saturated value.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#), signal2 elements to be multiplied by signal1 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.148.2.30 **NppStatus nppsMul\_8u\_Sfs** (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)

8-bit unsigned char signal times signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.148.2.31 **NppStatus nppsMul\_Low\_32s\_Sfs** (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal times signal, scale, then clamp to saturated value.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal2 elements to be multiplied by signal1 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.149 Sub

Sample by sample subtraction of the samples of two signals.

### Functions

- **NppStatus nppsSub\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal subtract signal, then clamp to saturated value.*
- **NppStatus nppsSub\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.*
- **NppStatus nppsSub\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.*
- **NppStatus nppsSub\_32sc\_Sfs** (const **Npp32sc** \*pSrc1, const **Npp32sc** \*pSrc2, **Npp32sc** \*pDst, int nLength, int nScaleFactor)

32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.

- **NppStatus nppsSub\_16s\_I** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength)  
16-bit signed short in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)  
32-bit floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)  
64-bit floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)  
32-bit complex floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)  
64-bit complex floating point in place signal subtract signal, then clamp to saturated value.
- **NppStatus nppsSub\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.
- **NppStatus nppsSub\_32sc\_ISfs** (const **Npp32sc** \*pSrc, **Npp32sc** \*pSrcDst, int nLength, int nScaleFactor)  
32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

### 7.149.1 Detailed Description

Sample by sample subtraction of the samples of two signals.

### 7.149.2 Function Documentation

#### 7.149.2.1 **NppStatus nppsSub\_16s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength)

16-bit signed short signal subtract signal, then clamp to saturated value.



**Parameters:**

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.149.2.2 NppStatus nppsSub\_16s32f (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)

16-bit signed short signal subtract 16-bit signed short signal, then clamp and convert to 32-bit floating point saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.  
*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.149.2.3 NppStatus nppsSub\_16s\_I (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*)

16-bit signed short in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.149.2.4 NppStatus nppsSub\_16s\_ISfs (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.149.2.5 NppStatus nppsSub\_16s\_Sfs (const Npp16s \* pSrc1, const Npp16s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)

16-bit signed short signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.149.2.6 NppStatus nppsSub\_16sc\_ISfs (const Npp16sc \* pSrc, Npp16sc \* pSrcDst, int nLength, int nScaleFactor)

16-bit complex signed short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.149.2.7 NppStatus nppsSub\_16sc\_Sfs (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, Npp16sc \* pDst, int nLength, int nScaleFactor)

16-bit signed complex short signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.2.8 NppStatus nppsSub\_16u\_ISfs (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#), signal1 elements to be subtracted from signal2 elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.2.9 NppStatus nppsSub\_16u\_Sfs (const Npp16u \* pSrc1, const Npp16u \* pSrc2, Npp16u \* pDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.2.10 NppStatus nppsSub\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, Npp32f \* pDst, int nLength)**

32-bit floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.2.11 NppStatus nppsSub\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.2.12 NppStatus nppsSub\_32fc (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).  
*pSrc2* [Source Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.2.13 NppStatus nppsSub\_32fc\_I (const Npp32fc \* pSrc, Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).  
*pSrcDst* [In-Place Signal Pointer](#). signal1 elements to be subtracted from signal2 elements  
*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.149.2.14 NppStatus nppsSub\_32s\_ISfs (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.15 NppStatus nppsSub\_32s\_Sfs (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.16 NppStatus nppsSub\_32sc\_ISfs (const Npp32sc \* *pSrc*, Npp32sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit complex signed integer in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.17 NppStatus nppsSub\_32sc\_Sfs (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, Npp32sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed complex integer signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.18 NppStatus nppsSub\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 elements to be subtracted from signal2 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.19 NppStatus nppsSub\_64f\_I (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer, signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.20 NppStatus nppsSub\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal1 elements to be subtracted from signal2 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.21 NppStatus nppsSub\_64fc\_I (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point in place signal subtract signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.22 NppStatus nppsSub\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal subtract signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 elements to be subtracted from signal2 elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.149.2.23** `NppStatus nppsSub_8u_Sfs (const Npp8u * pSrc1, const Npp8u * pSrc2, Npp8u * pDst, int nLength, int nScaleFactor)`

8-bit unsigned char signal subtract signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 elements to be subtracted from signal2 elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



## 7.150 Div

Sample by sample division of the samples of two signals.

### Functions

- **NppStatus nppsDiv\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_32s\_Sfs** (const **Npp32s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_16sc\_Sfs** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed complex short signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_32s16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp32s** \*pSrc2, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.*
- **NppStatus nppsDiv\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_32fc** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal divide signal, then clamp to saturated value.*
- **NppStatus nppsDiv\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.*
- **NppStatus nppsDiv\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.*

*16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_16sc\_ISfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)

*16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_32s\_ISfs** (const **Npp32s** \*pSrc, **Npp32s** \*pSrcDst, int nLength, int nScaleFactor)

*32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.*

- **NppStatus nppsDiv\_32f\_I** (const **Npp32f** \*pSrc, **Npp32f** \*pSrcDst, int nLength)

*32-bit floating point in place signal divide signal, then clamp to saturated value.*

- **NppStatus nppsDiv\_64f\_I** (const **Npp64f** \*pSrc, **Npp64f** \*pSrcDst, int nLength)

*64-bit floating point in place signal divide signal, then clamp to saturated value.*

- **NppStatus nppsDiv\_32fc\_I** (const **Npp32fc** \*pSrc, **Npp32fc** \*pSrcDst, int nLength)

*32-bit complex floating point in place signal divide signal, then clamp to saturated value.*

- **NppStatus nppsDiv\_64fc\_I** (const **Npp64fc** \*pSrc, **Npp64fc** \*pSrcDst, int nLength)

*64-bit complex floating point in place signal divide signal, then clamp to saturated value.*

## 7.150.1 Detailed Description

Sample by sample division of the samples of two signals.

## 7.150.2 Function Documentation

### 7.150.2.1 **NppStatus nppsDiv\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short in place signal divide signal, with scaling, then clamp to saturated value.

#### Parameters:

**pSrc** [Source Signal Pointer](#).

**pSrcDst** [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

**nLength** [Signal Length](#).

**nScaleFactor** [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.150.2.2 NppStatus nppsDiv\_16s\_Sfs (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed short signal divide signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.150.2.3 NppStatus nppsDiv\_16sc\_ISfs (const Npp16sc \* *pSrc*, Npp16sc \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

16-bit complex signed short in place signal divide signal, with scaling, then clamp to saturated value.

#### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

### 7.150.2.4 NppStatus nppsDiv\_16sc\_Sfs (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)

16-bit signed complex short signal divide signal, scale, then clamp to saturated value.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.150.2.5 NppStatus nppsDiv\_16u\_ISfs (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.150.2.6 NppStatus nppsDiv\_16u\_Sfs (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.150.2.7 NppStatus nppsDiv\_32f (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.150.2.8 NppStatus nppsDiv\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.150.2.9 NppStatus nppsDiv\_32fc (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, Npp32fc \* pDst, int nLength)**

32-bit complex floating point signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.150.2.10 NppStatus nppsDiv\_32fc\_I (const Npp32fc \* pSrc, Npp32fc \* pSrcDst, int nLength)**

32-bit complex floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.150.2.11 NppStatus nppsDiv\_32s16s\_Sfs (const Npp16s \* pSrc1, const Npp32s \* pSrc2, Npp16s \* pDst, int nLength, int nScaleFactor)**

32-bit signed integer signal divided by 16-bit signed short signal, scale, then clamp to 16-bit signed short saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.150.2.12 NppStatus nppsDiv\_32s\_ISfs (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.150.2.13 NppStatus nppsDiv\_32s\_Sfs (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)

32-bit signed integer signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.150.2.14 NppStatus nppsDiv\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, Npp64f \* *pDst*, int *nLength*)

64-bit floating point signal divide signal, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.150.2.15 NppStatus nppsDiv\_64f\_I (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)

64-bit floating point in place signal divide signal, then clamp to saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.150.2.16 NppStatus nppsDiv\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, Npp64fc \* *pDst*, int *nLength*)

64-bit complex floating point signal divide signal, then clamp to saturated value.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.150.2.17 NppStatus nppsDiv\_64fc\_I (const Npp64fc \* *pSrc*, Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point in place signal divide signal, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.150.2.18 NppStatus nppsDiv\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char in place signal divide signal, with scaling, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal1 divisor elements to be divided into signal2 dividend elements

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.150.2.19 NppStatus nppsDiv\_8u\_Sfs (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer, signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



## 7.151 Div\_Round

Sample by sample division of the samples of two signals with rounding.

### Functions

- **NppStatus nppsDiv\_Round\_8u\_Sfs** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*8-bit unsigned char signal divide signal, scale, then clamp to saturated value.*
- **NppStatus nppsDiv\_Round\_16u\_Sfs** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.*
- **NppStatus nppsDiv\_Round\_16s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, **Npp16s** \*pDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*16-bit signed short signal divide signal, scale, round, then clamp to saturated value.*
- **NppStatus nppsDiv\_Round\_8u\_ISfs** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.*
- **NppStatus nppsDiv\_Round\_16u\_ISfs** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.*
- **NppStatus nppsDiv\_Round\_16s\_ISfs** (const **Npp16s** \*pSrc, **Npp16s** \*pSrcDst, int nLength, **NppRoundMode** nRndMode, int nScaleFactor)  
*16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.*

### 7.151.1 Detailed Description

Sample by sample division of the samples of two signals with rounding.

### 7.151.2 Function Documentation

#### 7.151.2.1 NppStatus nppsDiv\_Round\_16s\_ISfs (const Npp16s \* pSrc, Npp16s \* pSrcDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)

16-bit signed short in place signal divide signal, with scaling, rounding then clamp to saturated value.

#### Parameters:

**pSrc** [Source Signal Pointer](#).

**pSrcDst** [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements  
**nLength** [Signal Length](#).

**nRndMode** various rounding modes.

**nScaleFactor** [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.2.2** `NppStatus nppsDiv_Round_16s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, Npp16s * pDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

16-bit signed short signal divide signal, scale, round, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.2.3** `NppStatus nppsDiv_Round_16u_ISfs (const Npp16u * pSrc, Npp16u * pSrcDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

16-bit unsigned short in place signal divide signal, with scaling, rounding then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.2.4** `NppStatus nppsDiv_Round_16u_Sfs (const Npp16u * pSrc1, const Npp16u * pSrc2, Npp16u * pDst, int nLength, NppRoundMode nRndMode, int nScaleFactor)`

16-bit unsigned short signal divide signal, scale, round, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.2.5 NppStatus nppsDiv\_Round\_8u\_ISfs (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

8-bit unsigned char in place signal divide signal, with scaling, rounding then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal1 divisor elements to be divided into signal2 dividend elements

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.151.2.6 NppStatus nppsDiv\_Round\_8u\_Sfs (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*, NppRoundMode *nRndMode*, int *nScaleFactor*)**

8-bit unsigned char signal divide signal, scale, then clamp to saturated value.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#), signal1 divisor elements to be divided into signal2 dividend elements.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nRndMode* various rounding modes.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.152 Abs

Absolute value of each sample of a signal.

### Functions

- **NppStatus nppsAbs\_16s** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal absolute value.*
- **NppStatus nppsAbs\_32s** (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength)  
*32-bit signed integer signal absolute value.*
- **NppStatus nppsAbs\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal absolute value.*
- **NppStatus nppsAbs\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal absolute value.*
- **NppStatus nppsAbs\_16s\_I** (**Npp16s** \*pSrcDst, int nLength)  
*16-bit signed short signal absolute value.*
- **NppStatus nppsAbs\_32s\_I** (**Npp32s** \*pSrcDst, int nLength)  
*32-bit signed integer signal absolute value.*
- **NppStatus nppsAbs\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal absolute value.*
- **NppStatus nppsAbs\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal absolute value.*

### 7.152.1 Detailed Description

Absolute value of each sample of a signal.

### 7.152.2 Function Documentation

#### 7.152.2.1 NppStatus nppsAbs\_16s (const Npp16s \*pSrc, Npp16s \*pDst, int nLength)

16-bit signed short signal absolute value.

#### Parameters:

**pSrc** Source Signal Pointer.  
**pDst** Destination Signal Pointer.  
**nLength** Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.2.2 NppStatus nppsAbs\_16s\_I (Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.2.3 NppStatus nppsAbs\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.2.4 NppStatus nppsAbs\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.2.5 NppStatus nppsAbs\_32s (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.2.6 NppStatus nppsAbs\_32s\_I (Npp32s \* *pSrcDst*, int *nLength*)**

32-bit signed integer signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.2.7 NppStatus nppsAbs\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal absolute value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.152.2.8 NppStatus nppsAbs\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal absolute value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.153 Sqr

Squares each sample of a signal.

### Functions

- **NppStatus nppsSqr\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal squared.*
- **NppStatus nppsSqr\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal squared.*
- **NppStatus nppsSqr\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal squared.*
- **NppStatus nppsSqr\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal squared.*
- **NppStatus nppsSqr\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point signal squared.*
- **NppStatus nppsSqr\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_8u\_ISfs** (**Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16u\_ISfs** (**Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal squared, scale, then clamp to saturated value.*

- **NppStatus nppsSqr\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal squared, scale, then clamp to saturated value.*
- **NppStatus nppsSqr\_16sc\_ISfs** (**Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal squared, scale, then clamp to saturated value.*

### 7.153.1 Detailed Description

Squares each sample of a signal.

### 7.153.2 Function Documentation

#### 7.153.2.1 NppStatus nppsSqr\_16s\_ISfs (Npp16s \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal squared, scale, then clamp to saturated value.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.153.2.2 NppStatus nppsSqr\_16s\_Sfs (const Npp16s \*pSrc, Npp16s \*pDst, int nLength, int nScaleFactor)

16-bit signed short signal squared, scale, then clamp to saturated value.

##### Parameters:

*pSrc* [Source Signal Pointer](#).  
*pDst* [Destination Signal Pointer](#).  
*nLength* [Signal Length](#).  
*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.153.2.3 NppStatus nppsSqr\_16sc\_ISfs (Npp16sc \*pSrcDst, int nLength, int nScaleFactor)

16-bit complex signed short signal squared, scale, then clamp to saturated value.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).



*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.4 NppStatus nppsSqr\_16sc\_Sfs (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit complex signed short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.5 NppStatus nppsSqr\_16u\_ISfs (Npp16u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.6 NppStatus nppsSqr\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.7 NppStatus nppsSqr\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.8 NppStatus nppsSqr\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.9 NppStatus nppsSqr\_32fc (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.10 NppStatus nppsSqr\_32fc\_I (Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.11 NppStatus nppsSqr\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.12 NppStatus nppsSqr\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.13 NppStatus nppsSqr\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex floating point signal squared.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.14 NppStatus nppsSqr\_64fc\_I (Npp64fc \* *pSrcDst*, int *nLength*)**

64-bit complex floating point signal squared.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.15 NppStatus nppsSqr\_8u\_ISfs (Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.153.2.16 NppStatus nppsSqr\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal squared, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.154 Sqrt

Square root of each sample of a signal.

### Functions

- **NppStatus nppsSqrt\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal square root.*
- **NppStatus nppsSqrt\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal square root.*
- **NppStatus nppsSqrt\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength)  
*32-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength)  
*64-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal square root.*
- **NppStatus nppsSqrt\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal square root.*
- **NppStatus nppsSqrt\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength)  
*32-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength)  
*64-bit complex floating point signal square root.*
- **NppStatus nppsSqrt\_8u\_Sfs** (const **Npp8u** \*pSrc, **Npp8u** \*pDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16u\_Sfs** (const **Npp16u** \*pSrc, **Npp16u** \*pDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_64s\_Sfs** (const **Npp64s** \*pSrc, **Npp64s** \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_32s16s\_Sfs** (const **Npp32s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.*

- **NppStatus nppsSqrt\_64s16s\_Sfs** (const **Npp64s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.*
- **NppStatus nppsSqrt\_8u\_ISfs** (**Npp8u** \*pSrcDst, int nLength, int nScaleFactor)  
*8-bit unsigned char signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16u\_ISfs** (**Npp16u** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit unsigned short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_16sc\_ISfs** (**Npp16sc** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit complex signed short signal square root, scale, then clamp to saturated value.*
- **NppStatus nppsSqrt\_64s\_ISfs** (**Npp64s** \*pSrcDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal square root, scale, then clamp to saturated value.*

### 7.154.1 Detailed Description

Square root of each sample of a signal.

### 7.154.2 Function Documentation

#### 7.154.2.1 **NppStatus nppsSqrt\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal square root, scale, then clamp to saturated value.

##### Parameters:

**pSrcDst** In-Place Signal Pointer.

**nLength** Signal Length.

**nScaleFactor** Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.154.2.2 **NppStatus nppsSqrt\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)

16-bit signed short signal square root, scale, then clamp to saturated value.

##### Parameters:

**pSrc** Source Signal Pointer.

*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.3 NppStatus nppsSqrt\_16sc\_ISfs (Npp16sc \* pSrcDst, int nLength, int nScaleFactor)**

16-bit complex signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.4 NppStatus nppsSqrt\_16sc\_Sfs (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength, int nScaleFactor)**

16-bit complex signed short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.5 NppStatus nppsSqrt\_16u\_ISfs (Npp16u \* pSrcDst, int nLength, int nScaleFactor)**

16-bit unsigned short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.6 NppStatus nppsSqrt\_16u\_Sfs (const Npp16u \* *pSrc*, Npp16u \* *pDst*, int *nLength*, int *nScaleFactor*)**

16-bit unsigned short signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.7 NppStatus nppsSqrt\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

32-bit floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.8 NppStatus nppsSqrt\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.9 NppStatus nppsSqrt\_32fc (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*)**

32-bit complex floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.



*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.10 NppStatus nppsSqrt\_32fc\_I (Npp32fc \* *pSrcDst*, int *nLength*)**

32-bit complex floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.11 NppStatus nppsSqrt\_32s16s\_Sfs (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.12 NppStatus nppsSqrt\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.13 NppStatus nppsSqrt\_64f\_I (Npp64f \* pSrcDst, int nLength)**

64-bit floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.14 NppStatus nppsSqrt\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength)**

64-bit complex floating point signal square root.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.15 NppStatus nppsSqrt\_64fc\_I (Npp64fc \* pSrcDst, int nLength)**

64-bit complex floating point signal square root.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.16 NppStatus nppsSqrt\_64s16s\_Sfs (const Npp64s \* pSrc, Npp16s \* pDst, int nLength, int nScaleFactor)**

64-bit signed integer signal square root, scale, then clamp to 16-bit signed integer saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.154.2.17 NppStatus nppsSqrt\_64s\_ISfs (Npp64s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.154.2.18 NppStatus nppsSqrt\_64s\_Sfs (const Npp64s \* *pSrc*, Npp64s \* *pDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.154.2.19 NppStatus nppsSqrt\_8u\_ISfs (Npp8u \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.154.2.20 NppStatus nppsSqrt\_8u\_Sfs (const Npp8u \* *pSrc*, Npp8u \* *pDst*, int *nLength*, int *nScaleFactor*)**

8-bit unsigned char signal square root, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.155 Cubrt

Cube root of each sample of a signal.

### Functions

- **NppStatus nppsCubrt\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal cube root.*
- **NppStatus nppsCubrt\_32s16s\_Sfs** (const **Npp32s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.*

### 7.155.1 Detailed Description

Cube root of each sample of a signal.

### 7.155.2 Function Documentation

#### 7.155.2.1 NppStatus nppsCubrt\_32f (const Npp32f \*pSrc, Npp32f \*pDst, int nLength)

32-bit floating point signal cube root.

##### Parameters:

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.155.2.2 NppStatus nppsCubrt\_32s16s\_Sfs (const Npp32s \*pSrc, Npp16s \*pDst, int nLength, int nScaleFactor)

32-bit signed integer signal cube root, scale, then clamp to 16-bit signed integer saturated value.

##### Parameters:

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

## 7.156 Exp

E raised to the power of each sample of a signal.

### Functions

- **NppStatus nppsExp\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength)  
*32-bit floating point signal exponent.*
- **NppStatus nppsExp\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*64-bit floating point signal exponent.*
- **NppStatus nppsExp\_32f64f** (const **Npp32f** \*pSrc, **Npp64f** \*pDst, int nLength)  
*32-bit floating point signal exponent with 64-bit floating point result.*
- **NppStatus nppsExp\_32f\_I** (**Npp32f** \*pSrcDst, int nLength)  
*32-bit floating point signal exponent.*
- **NppStatus nppsExp\_64f\_I** (**Npp64f** \*pSrcDst, int nLength)  
*64-bit floating point signal exponent.*
- **NppStatus nppsExp\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_64s\_Sfs** (const **Npp64s** \*pSrc, **Npp64s** \*pDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_16s\_ISfs** (**Npp16s** \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_32s\_ISfs** (**Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal exponent, scale, then clamp to saturated value.*
- **NppStatus nppsExp\_64s\_ISfs** (**Npp64s** \*pSrcDst, int nLength, int nScaleFactor)  
*64-bit signed integer signal exponent, scale, then clamp to saturated value.*

### 7.156.1 Detailed Description

E raised to the power of each sample of a signal.

### 7.156.2 Function Documentation

#### 7.156.2.1 NppStatus nppsExp\_16s\_ISfs (Npp16s \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.2 NppStatus nppsExp\_16s\_Sfs (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.3 NppStatus nppsExp\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal exponent.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.4 NppStatus nppsExp\_32f64f (const Npp32f \* pSrc, Npp64f \* pDst, int nLength)**

32-bit floating point signal exponent with 64-bit floating point result.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.5 NppStatus nppsExp\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point signal exponent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.6 NppStatus nppsExp\_32s\_ISfs (Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.7 NppStatus nppsExp\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.8 NppStatus nppsExp\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal exponent.

**Parameters:**

*pSrc* Source Signal Pointer.



*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.9 NppStatus nppsExp\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal exponent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.10 NppStatus nppsExp\_64s\_ISfs (Npp64s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.156.2.11 NppStatus nppsExp\_64s\_Sfs (const Npp64s \* *pSrc*, Npp64s \* *pDst*, int *nLength*, int *nScaleFactor*)**

64-bit signed integer signal exponent, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.157 Ln

Natural logarithm of each sample of a signal.

### Functions

- `NppStatus nppsLn_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_64f32f` (const `Npp64f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*64-bit floating point signal natural logarithm with 32-bit floating point result.*
- `NppStatus nppsLn_32f_I` (`Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_64f_I` (`Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point signal natural logarithm.*
- `NppStatus nppsLn_16s_Sfs` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*16-bit signed short signal natural logarithm, scale, then clamp to saturated value.*
- `NppStatus nppsLn_32s_Sfs` (const `Npp32s` \*pSrc, `Npp32s` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.*
- `NppStatus nppsLn_32s16s_Sfs` (const `Npp32s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.*
- `NppStatus nppsLn_16s_ISfs` (`Npp16s` \*pSrcDst, int nLength, int nScaleFactor)  
*16-bit signed short signal natural logarithm, scale, then clamp to saturated value.*
- `NppStatus nppsLn_32s_ISfs` (`Npp32s` \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.*

### 7.157.1 Detailed Description

Natural logarithm of each sample of a signal.

### 7.157.2 Function Documentation

#### 7.157.2.1 `NppStatus nppsLn_16s_ISfs` (`Npp16s` \*pSrcDst, int nLength, int nScaleFactor)

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.2 NppStatus nppsLn\_16s\_Sfs (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, int nScaleFactor)**

16-bit signed short signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.3 NppStatus nppsLn\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit floating point signal natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.4 NppStatus nppsLn\_32f\_I (Npp32f \* pSrcDst, int nLength)**

32-bit floating point signal natural logarithm.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.5 NppStatus nppsLn\_32s16s\_Sfs (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal natural logarithm, scale, then clamp to 16-bit signed short saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.6 NppStatus nppsLn\_32s\_ISfs (Npp32s \* *pSrcDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.7 NppStatus nppsLn\_32s\_Sfs (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*, int *nScaleFactor*)**

32-bit signed integer signal natural logarithm, scale, then clamp to saturated value.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.  
*nScaleFactor* Integer Result Scaling.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.8 NppStatus nppsLn\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal natural logarithm.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.9 NppStatus nppsLn\_64f32f (const Npp64f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)**

64-bit floating point signal natural logarithm with 32-bit floating point result.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.157.2.10 NppStatus nppsLn\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal natural logarithm.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.158 10Log10

Ten times the decimal logarithm of each sample of a signal.

### Functions

- **NppStatus npps10Log10\_32s\_Sfs** (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.*
- **NppStatus npps10Log10\_32s\_ISfs** (**Npp32s** \*pSrcDst, int nLength, int nScaleFactor)  
*32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.*

### 7.158.1 Detailed Description

Ten times the decimal logarithm of each sample of a signal.

### 7.158.2 Function Documentation

#### 7.158.2.1 NppStatus npps10Log10\_32s\_ISfs (**Npp32s** \*pSrcDst, int nLength, int nScaleFactor)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

##### Parameters:

**pSrcDst** In-Place Signal Pointer.  
**nLength** Signal Length.  
**nScaleFactor** Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.158.2.2 NppStatus npps10Log10\_32s\_Sfs (const **Npp32s** \*pSrc, **Npp32s** \*pDst, int nLength, int nScaleFactor)

32-bit signed integer signal 10 times base 10 logarithm, scale, then clamp to saturated value.

##### Parameters:

**pSrc** Source Signal Pointer.  
**pDst** Destination Signal Pointer.  
**nLength** Signal Length.  
**nScaleFactor** Integer Result Scaling.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

## 7.159 SumLn

Sums up the natural logarithm of each sample of a signal.

### Functions

- **NppStatus nppsSumLnGetBufferSize\_32f** (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32f SumLn.*
- **NppStatus nppsSumLn\_32f** (const Npp32f \*pSrc, int nLength, Npp32f \*pDst, Npp8u \*pDeviceBuffer)  
*32-bit floating point signal sum natural logarithm.*
- **NppStatus nppsSumLnGetBufferSize\_64f** (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 64f SumLn.*
- **NppStatus nppsSumLn\_64f** (const Npp64f \*pSrc, int nLength, Npp64f \*pDst, Npp8u \*pDeviceBuffer)  
*64-bit floating point signal sum natural logarithm.*
- **NppStatus nppsSumLnGetBufferSize\_32f64f** (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 32f64f SumLn.*
- **NppStatus nppsSumLn\_32f64f** (const Npp32f \*pSrc, int nLength, Npp64f \*pDst, Npp8u \*pDeviceBuffer)  
*32-bit floating point input, 64-bit floating point output signal sum natural logarithm.*
- **NppStatus nppsSumLnGetBufferSize\_16s32f** (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for 16s32f SumLn.*
- **NppStatus nppsSumLn\_16s32f** (const Npp16s \*pSrc, int nLength, Npp32f \*pDst, Npp8u \*pDeviceBuffer)  
*16-bit signed short integer input, 32-bit floating point output signal sum natural logarithm.*

### 7.159.1 Detailed Description

Sums up the natural logarithm of each sample of a signal.

### 7.159.2 Function Documentation

#### 7.159.2.1 NppStatus nppsSumLn\_16s32f (const Npp16s \*pSrc, int nLength, Npp32f \*pDst, Npp8u \*pDeviceBuffer)

16-bit signed short integer input, 32-bit floating point output signal sum natural logarithm.

#### Parameters:

*pSrc* Source Signal Pointer.

*nLength* [Signal Length](#).

*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.159.2.2 NppStatus nppsSumLn\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pDst, Npp8u \* pDeviceBuffer)**

32-bit floating point signal sum natural logarithm.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.159.2.3 NppStatus nppsSumLn\_32f64f (const Npp32f \* pSrc, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

32-bit floating point input, 64-bit floating point output signal sum natural logarithm.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.159.2.4 NppStatus nppsSumLn\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

64-bit floating point signal sum natural logarithm.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).



*pDst* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.159.2.5 NppStatus nppsSumLnGetBufferSize\_16s32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 16s32f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.159.2.6 NppStatus nppsSumLnGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.159.2.7 NppStatus nppsSumLnGetBufferSize\_32f64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 32f64f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_32f64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.159.2.8 NppStatus nppsSumLnGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for 64f SumLn.

This primitive provides the correct buffer size for nppsSumLn\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.160 Arctan

Inverse tangent of each sample of a signal.

### Functions

- `NppStatus nppsArctan_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit floating point signal inverse tangent.*
- `NppStatus nppsArctan_64f` (const `Npp64f` \*pSrc, `Npp64f` \*pDst, int nLength)  
*64-bit floating point signal inverse tangent.*
- `NppStatus nppsArctan_32f_I` (`Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point signal inverse tangent.*
- `NppStatus nppsArctan_64f_I` (`Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point signal inverse tangent.*

### 7.160.1 Detailed Description

Inverse tangent of each sample of a signal.

### 7.160.2 Function Documentation

#### 7.160.2.1 `NppStatus nppsArctan_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)

32-bit floating point signal inverse tangent.

##### Parameters:

*pSrc* Source Signal Pointer.  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.160.2.2 `NppStatus nppsArctan_32f_I` (`Npp32f` \*pSrcDst, int nLength)

32-bit floating point signal inverse tangent.

##### Parameters:

*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.160.2.3 NppStatus nppsArctan\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)**

64-bit floating point signal inverse tangent.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.160.2.4 NppStatus nppsArctan\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point signal inverse tangent.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.161 Normalize

Normalize each sample of a real or complex signal using offset and division operations.

### Functions

- **NppStatus nppsNormalize\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength, **Npp32f** vSub, **Npp32f** vDiv)  
*32-bit floating point signal normalize.*
- **NppStatus nppsNormalize\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength, **Npp32fc** vSub, **Npp32f** vDiv)  
*32-bit complex floating point signal normalize.*
- **NppStatus nppsNormalize\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength, **Npp64f** vSub, **Npp64f** vDiv)  
*64-bit floating point signal normalize.*
- **NppStatus nppsNormalize\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength, **Npp64fc** vSub, **Npp64f** vDiv)  
*64-bit complex floating point signal normalize.*
- **NppStatus nppsNormalize\_16s\_Sfs** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** vSub, int vDiv, int nScaleFactor)  
*16-bit signed short signal normalize, scale, then clamp to saturated value.*
- **NppStatus nppsNormalize\_16sc\_Sfs** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, **Npp16sc** vSub, int vDiv, int nScaleFactor)  
*16-bit complex signed short signal normalize, scale, then clamp to saturated value.*

### 7.161.1 Detailed Description

Normalize each sample of a real or complex signal using offset and division operations.

### 7.161.2 Function Documentation

#### 7.161.2.1 NppStatus nppsNormalize\_16s\_Sfs (const Npp16s \* pSrc, Npp16s \* pDst, int nLength, Npp16s vSub, int vDiv, int nScaleFactor)

16-bit signed short signal normalize, scale, then clamp to saturated value.

#### Parameters:

**pSrc** Source Signal Pointer.

**pDst** Destination Signal Pointer.

**nLength** Signal Length.

**vSub** value subtracted from each signal element before division

**vDiv** divisor of post-subtracted signal element dividend

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.161.2.2** `NppStatus nppsNormalize_16sc_Sfs (const Npp16sc * pSrc, Npp16sc * pDst, int nLength, Npp16sc vSub, int vDiv, int nScaleFactor)`

16-bit complex signed short signal normalize, scale, then clamp to saturated value.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.161.2.3** `NppStatus nppsNormalize_32f (const Npp32f * pSrc, Npp32f * pDst, int nLength, Npp32f vSub, Npp32f vDiv)`

32-bit floating point signal normalize.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.161.2.4** `NppStatus nppsNormalize_32fc (const Npp32fc * pSrc, Npp32fc * pDst, int nLength, Npp32fc vSub, Npp32fc vDiv)`

32-bit complex floating point signal normalize.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.161.2.5** `NppStatus nppsNormalize_64f (const Npp64f * pSrc, Npp64f * pDst, int nLength, Npp64f vSub, Npp64f vDiv)`

64-bit floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.161.2.6** `NppStatus nppsNormalize_64fc (const Npp64fc * pSrc, Npp64fc * pDst, int nLength, Npp64fc vSub, Npp64fc vDiv)`

64-bit complex floating point signal normalize.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*vSub* value subtracted from each signal element before division

*vDiv* divisor of post-subtracted signal element dividend

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.162 Cauchy, CauchyD, and CauchyDD2

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

### Functions

- **NppStatus nppsCauchy\_32f\_I** (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nParam)  
*32-bit floating point signal Cauchy error calculation.*
- **NppStatus nppsCauchyD\_32f\_I** (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nParam)  
*32-bit floating point signal Cauchy first derivative.*
- **NppStatus nppsCauchyDD2\_32f\_I** (**Npp32f** \*pSrcDst, **Npp32f** \*pD2FVal, int nLength, **Npp32f** nParam)  
*32-bit floating point signal Cauchy first and second derivatives.*

### 7.162.1 Detailed Description

Determine Cauchy robust error function and its first and second derivatives for each sample of a signal.

### 7.162.2 Function Documentation

#### 7.162.2.1 NppStatus nppsCauchy\_32f\_I (Npp32f \*pSrcDst, int nLength, Npp32f nParam)

32-bit floating point signal Cauchy error calculation.

##### Parameters:

**pSrcDst** [In-Place Signal Pointer](#).  
**nLength** [Signal Length](#).  
**nParam** constant used in Cauchy formula

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.162.2.2 NppStatus nppsCauchyD\_32f\_I (Npp32f \*pSrcDst, int nLength, Npp32f nParam)

32-bit floating point signal Cauchy first derivative.

##### Parameters:

**pSrcDst** [In-Place Signal Pointer](#).  
**nLength** [Signal Length](#).  
**nParam** constant used in Cauchy formula

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.162.2.3 NppStatus nppsCauchyDD2\_32f\_I (Npp32f \* *pSrcDst*, Npp32f \* *pD2FVal*, int *nLength*, Npp32f *nParam*)**

32-bit floating point signal Cauchy first and second derivatives.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*pD2FVal* [Source Signal Pointer](#). This signal contains the second derivative of the source signal.

*nLength* [Signal Length](#).

*nParam* constant used in Cauchy formula

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.163 Logical And Shift Operations

### Modules

- [AndC](#)

*Bitwise AND of a constant and each sample of a signal.*

- [And](#)

*Sample by sample bitwise AND of samples from two signals.*

- [OrC](#)

*Bitwise OR of a constant and each sample of a signal.*

- [Or](#)

*Sample by sample bitwise OR of the samples from two signals.*

- [XorC](#)

*Bitwise XOR of a constant and each sample of a signal.*

- [Xor](#)

*Sample by sample bitwise XOR of the samples from two signals.*

- [Not](#)

*Bitwise NOT of each sample of a signal.*

- [LShiftC](#)

*Left shifts the bits of each sample of a signal by a constant amount.*

- [RShiftC](#)

*Right shifts the bits of each sample of a signal by a constant amount.*

## 7.164 AndC

Bitwise AND of a constant and each sample of a signal.

### Functions

- **NppStatus nppsAndC\_8u** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal and with constant.*
- **NppStatus nppsAndC\_16u** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal and with constant.*
- **NppStatus nppsAndC\_32u** (const **Npp32u** \*pSrc, **Npp32u** nValue, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal and with constant.*
- **NppStatus nppsAndC\_8u\_I** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal and with constant.*
- **NppStatus nppsAndC\_16u\_I** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal and with constant.*
- **NppStatus nppsAndC\_32u\_I** (**Npp32u** nValue, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal and with constant.*

### 7.164.1 Detailed Description

Bitwise AND of a constant and each sample of a signal.

### 7.164.2 Function Documentation

#### 7.164.2.1 NppStatus nppsAndC\_16u (const Npp16u \*pSrc, Npp16u nValue, Npp16u \*pDst, int nLength)

16-bit unsigned short signal and with constant.

#### Parameters:

- pSrc** Source Signal Pointer.  
**nValue** Constant value to be anded with each vector element  
**pDst** Destination Signal Pointer.  
**nLength** Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.164.2.2 NppStatus nppsAndC\_16u\_I (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal and with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.164.2.3 NppStatus nppsAndC\_32u (const Npp32u \* *pSrc*, Npp32u *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal and with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be added with each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.164.2.4 NppStatus nppsAndC\_32u\_I (Npp32u *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal and with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be added with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.164.2.5 NppStatus nppsAndC\_8u (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal and with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be anded with each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.164.2.6 NppStatus nppsAndC\_8u\_I (Npp8u nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal and with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be anded with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.165 And

Sample by sample bitwise AND of samples from two signals.

### Functions

- **NppStatus nppsAnd\_8u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal and with signal.*
- **NppStatus nppsAnd\_16u** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal and with signal.*
- **NppStatus nppsAnd\_32u** (const **Npp32u** \*pSrc1, const **Npp32u** \*pSrc2, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal and with signal.*
- **NppStatus nppsAnd\_8u\_I** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal and with signal.*
- **NppStatus nppsAnd\_16u\_I** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal and with signal.*
- **NppStatus nppsAnd\_32u\_I** (const **Npp32u** \*pSrc, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal and with signal.*

### 7.165.1 Detailed Description

Sample by sample bitwise AND of samples from two signals.

### 7.165.2 Function Documentation

#### 7.165.2.1 **NppStatus nppsAnd\_16u** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength)

16-bit unsigned short signal and with signal.

#### Parameters:

- pSrc1** Source Signal Pointer.  
**pSrc2** Source Signal Pointer. signal2 elements to be anded with signal1 elements  
**pDst** Destination Signal Pointer.  
**nLength** Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.165.2.2 NppStatus nppsAnd\_16u\_I (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal and with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.165.2.3 NppStatus nppsAnd\_32u (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal and with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be anded with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.165.2.4 NppStatus nppsAnd\_32u\_I (const Npp32u \* *pSrc*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned integer in place signal and with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.165.2.5 NppStatus nppsAnd\_8u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal and with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be anded with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.165.2.6 NppStatus nppsAnd\_8u\_I (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal and with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be anded with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



## 7.166 OrC

Bitwise OR of a constant and each sample of a signal.

### Functions

- **NppStatus nppsOrC\_8u** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal or with constant.*
- **NppStatus nppsOrC\_16u** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal or with constant.*
- **NppStatus nppsOrC\_32u** (const **Npp32u** \*pSrc, **Npp32u** nValue, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal or with constant.*
- **NppStatus nppsOrC\_8u\_I** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal or with constant.*
- **NppStatus nppsOrC\_16u\_I** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal or with constant.*
- **NppStatus nppsOrC\_32u\_I** (**Npp32u** nValue, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal or with constant.*

### 7.166.1 Detailed Description

Bitwise OR of a constant and each sample of a signal.

### 7.166.2 Function Documentation

#### 7.166.2.1 NppStatus nppsOrC\_16u (const Npp16u \*pSrc, Npp16u nValue, Npp16u \*pDst, int nLength)

16-bit unsigned short signal or with constant.

#### Parameters:

- pSrc** [Source Signal Pointer](#).  
**nValue** Constant value to be ored with each vector element  
**pDst** [Destination Signal Pointer](#).  
**nLength** [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.166.2.2 NppStatus nppsOrC\_16u\_I (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal or with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be ored with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.166.2.3 NppStatus nppsOrC\_32u (const Npp32u \* *pSrc*, Npp32u *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal or with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be ored with each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.166.2.4 NppStatus nppsOrC\_32u\_I (Npp32u *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal or with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be ored with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.166.2.5 NppStatus nppsOrC\_8u (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal or with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.166.2.6 NppStatus nppsOrC\_8u\_I (Npp8u *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.167 Or

Sample by sample bitwise OR of the samples from two signals.

### Functions

- **NppStatus nppsOr\_8u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal or with signal.*
- **NppStatus nppsOr\_16u** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal or with signal.*
- **NppStatus nppsOr\_32u** (const **Npp32u** \*pSrc1, const **Npp32u** \*pSrc2, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal or with signal.*
- **NppStatus nppsOr\_8u\_I** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal or with signal.*
- **NppStatus nppsOr\_16u\_I** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal or with signal.*
- **NppStatus nppsOr\_32u\_I** (const **Npp32u** \*pSrc, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal or with signal.*

### 7.167.1 Detailed Description

Sample by sample bitwise OR of the samples from two signals.

### 7.167.2 Function Documentation

#### 7.167.2.1 NppStatus nppsOr\_16u (const Npp16u \*pSrc1, const Npp16u \*pSrc2, Npp16u \*pDst, int nLength)

16-bit unsigned short signal or with signal.

#### Parameters:

- pSrc1** [Source Signal Pointer](#).  
**pSrc2** [Source Signal Pointer](#). signal2 elements to be ored with signal1 elements  
**pDst** [Destination Signal Pointer](#).  
**nLength** [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.167.2.2 NppStatus nppsOr\_16u\_I (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.167.2.3 NppStatus nppsOr\_32u (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be ored with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.167.2.4 NppStatus nppsOr\_32u\_I (const Npp32u \* *pSrc*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned integer in place signal or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.167.2.5 NppStatus nppsOr\_8u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal or with signal.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer. signal2 elements to be ored with signal1 elements

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.167.2.6 NppStatus nppsOr\_8u\_I (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal or with signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer. signal2 elements to be ored with signal1 elements

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.168 XorC

Bitwise XOR of a constant and each sample of a signal.

### Functions

- **NppStatus nppsXorC\_8u** (const **Npp8u** \*pSrc, **Npp8u** nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal exclusive or with constant.*
- **NppStatus nppsXorC\_16u** (const **Npp16u** \*pSrc, **Npp16u** nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal exclusive or with constant.*
- **NppStatus nppsXorC\_32u** (const **Npp32u** \*pSrc, **Npp32u** nValue, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal exclusive or with constant.*
- **NppStatus nppsXorC\_8u\_I** (**Npp8u** nValue, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal exclusive or with constant.*
- **NppStatus nppsXorC\_16u\_I** (**Npp16u** nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal exclusive or with constant.*
- **NppStatus nppsXorC\_32u\_I** (**Npp32u** nValue, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal exclusive or with constant.*

### 7.168.1 Detailed Description

Bitwise XOR of a constant and each sample of a signal.

### 7.168.2 Function Documentation

#### 7.168.2.1 NppStatus nppsXorC\_16u (const Npp16u \*pSrc, Npp16u nValue, Npp16u \*pDst, int nLength)

16-bit unsigned short signal exclusive or with constant.

#### Parameters:

**pSrc** [Source Signal Pointer](#).

**nValue** Constant value to be exclusive ored with each vector element

**pDst** [Destination Signal Pointer](#).

**nLength** [Signal Length](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.168.2.2 NppStatus nppsXorC\_16u\_I (Npp16u *nValue*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be exclusive ored with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.168.2.3 NppStatus nppsXorC\_32u (const Npp32u \* *pSrc*, Npp32u *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal exclusive or with constant.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nValue* Constant value to be exclusive ored with each vector element

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.168.2.4 NppStatus nppsXorC\_32u\_I (Npp32u *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nValue* Constant value to be exclusive ored with each vector element

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.168.2.5 NppStatus nppsXorC\_8u (const Npp8u \* *pSrc*, Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal exclusive or with constant.



**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.168.2.6 NppStatus nppsXorC\_8u\_I (Npp8u nValue, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal exclusive or with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be exclusive ored with each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.169 Xor

Sample by sample bitwise XOR of the samples from two signals.

### Functions

- **NppStatus nppsXor\_8u** (const **Npp8u** \*pSrc1, const **Npp8u** \*pSrc2, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal exclusive or with signal.*
- **NppStatus nppsXor\_16u** (const **Npp16u** \*pSrc1, const **Npp16u** \*pSrc2, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal exclusive or with signal.*
- **NppStatus nppsXor\_32u** (const **Npp32u** \*pSrc1, const **Npp32u** \*pSrc2, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal exclusive or with signal.*
- **NppStatus nppsXor\_8u\_I** (const **Npp8u** \*pSrc, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal exclusive or with signal.*
- **NppStatus nppsXor\_16u\_I** (const **Npp16u** \*pSrc, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal exclusive or with signal.*
- **NppStatus nppsXor\_32u\_I** (const **Npp32u** \*pSrc, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned integer in place signal exclusive or with signal.*

### 7.169.1 Detailed Description

Sample by sample bitwise XOR of the samples from two signals.

### 7.169.2 Function Documentation

#### 7.169.2.1 NppStatus nppsXor\_16u (const Npp16u \*pSrc1, const Npp16u \*pSrc2, Npp16u \*pDst, int nLength)

16-bit unsigned short signal exclusive or with signal.

#### Parameters:

- pSrc1** Source Signal Pointer.  
**pSrc2** Source Signal Pointer. signal2 elements to be exclusive ored with signal1 elements  
**pDst** Destination Signal Pointer.  
**nLength** Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.169.2.2 NppStatus nppsXor\_16u\_I (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short in place signal exclusive or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.169.2.3 NppStatus nppsXor\_32u (const Npp32u \* pSrc1, const Npp32u \* pSrc2, Npp32u \* pDst, int nLength)**

32-bit unsigned integer signal exclusive or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.169.2.4 NppStatus nppsXor\_32u\_I (const Npp32u \* pSrc, Npp32u \* pSrcDst, int nLength)**

32-bit unsigned integer in place signal exclusive or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.169.2.5 NppStatus nppsXor\_8u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, Npp8u \* pDst, int nLength)**

8-bit unsigned char signal exclusive or with signal.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.169.2.6 NppStatus nppsXor\_8u\_I (const Npp8u \* pSrc, Npp8u \* pSrcDst, int nLength)**

8-bit unsigned char in place signal exclusive or with signal.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pSrcDst* [In-Place Signal Pointer](#). signal2 elements to be exclusive ored with signal1 elements

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.170 Not

Bitwise NOT of each sample of a signal.

### Functions

- **NppStatus nppsNot\_8u** (const **Npp8u** \*pSrc, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char not signal.*
- **NppStatus nppsNot\_16u** (const **Npp16u** \*pSrc, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short not signal.*
- **NppStatus nppsNot\_32u** (const **Npp32u** \*pSrc, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer not signal.*
- **NppStatus nppsNot\_8u\_I** (**Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place not signal.*
- **NppStatus nppsNot\_16u\_I** (**Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place not signal.*
- **NppStatus nppsNot\_32u\_I** (**Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place not signal.*

### 7.170.1 Detailed Description

Bitwise NOT of each sample of a signal.

### 7.170.2 Function Documentation

#### 7.170.2.1 NppStatus nppsNot\_16u (const Npp16u \*pSrc, Npp16u \*pDst, int nLength)

16-bit unsigned short not signal.

#### Parameters:

**pSrc** Source Signal Pointer.

**pDst** Destination Signal Pointer.

**nLength** Signal Length.

#### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.170.2.2 NppStatus nppsNot\_16u\_I (Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.170.2.3 NppStatus nppsNot\_32u (const Npp32u \* *pSrc*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer not signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.170.2.4 NppStatus nppsNot\_32u\_I (Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.170.2.5 NppStatus nppsNot\_8u (const Npp8u \* *pSrc*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char not signal.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.170.2.6 NppStatus nppsNot\_8u\_I (Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place not signal.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.171 LShiftC

Left shifts the bits of each sample of a signal by a constant amount.

### Functions

- **NppStatus nppsLShiftC\_8u** (const **Npp8u** \*pSrc, int nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal left shift with constant.*
- **NppStatus nppsLShiftC\_16u** (const **Npp16u** \*pSrc, int nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal left shift with constant.*
- **NppStatus nppsLShiftC\_16s** (const **Npp16s** \*pSrc, int nValue, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal left shift with constant.*
- **NppStatus nppsLShiftC\_32u** (const **Npp32u** \*pSrc, int nValue, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal left shift with constant.*
- **NppStatus nppsLShiftC\_32s** (const **Npp32s** \*pSrc, int nValue, **Npp32s** \*pDst, int nLength)  
*32-bit signed integer signal left shift with constant.*
- **NppStatus nppsLShiftC\_8u\_I** (int nValue, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_16u\_I** (int nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_16s\_I** (int nValue, **Npp16s** \*pSrcDst, int nLength)  
*16-bit signed short in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_32u\_I** (int nValue, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal left shift with constant.*
- **NppStatus nppsLShiftC\_32s\_I** (int nValue, **Npp32s** \*pSrcDst, int nLength)  
*32-bit signed signed integer in place signal left shift with constant.*

### 7.171.1 Detailed Description

Left shifts the bits of each sample of a signal by a constant amount.

### 7.171.2 Function Documentation

#### 7.171.2.1 **NppStatus nppsLShiftC\_16s** (const **Npp16s** \*pSrc, int nValue, **Npp16s** \*pDst, int nLength)

16-bit signed short signal left shift with constant.



**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.2 NppStatus nppsLShiftC\_16s\_I (int *nValue*, Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.3 NppStatus nppsLShiftC\_16u (const Npp16u \* *pSrc*, int *nValue*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.4 NppStatus nppsLShiftC\_16u\_I (int *nValue*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.5 NppStatus nppsLShiftC\_32s (const Npp32s \* *pSrc*, int *nValue*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.6 NppStatus nppsLShiftC\_32s\_I (int *nValue*, Npp32s \* *pSrcDst*, int *nLength*)**

32-bit signed integer in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.7 NppStatus nppsLShiftC\_32u (const Npp32u \* *pSrc*, int *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.8 NppStatus nppsLShiftC\_32u\_I (int *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.9 NppStatus nppsLShiftC\_8u (const Npp8u \* *pSrc*, int *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal left shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.171.2.10 NppStatus nppsLShiftC\_8u\_I (int *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal left shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to left shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.172 RShiftC

Right shifts the bits of each sample of a signal by a constant amount.

### Functions

- **NppStatus nppsRShiftC\_8u** (const **Npp8u** \*pSrc, int nValue, **Npp8u** \*pDst, int nLength)  
*8-bit unsigned char signal right shift with constant.*
- **NppStatus nppsRShiftC\_16u** (const **Npp16u** \*pSrc, int nValue, **Npp16u** \*pDst, int nLength)  
*16-bit unsigned short signal right shift with constant.*
- **NppStatus nppsRShiftC\_16s** (const **Npp16s** \*pSrc, int nValue, **Npp16s** \*pDst, int nLength)  
*16-bit signed short signal right shift with constant.*
- **NppStatus nppsRShiftC\_32u** (const **Npp32u** \*pSrc, int nValue, **Npp32u** \*pDst, int nLength)  
*32-bit unsigned integer signal right shift with constant.*
- **NppStatus nppsRShiftC\_32s** (const **Npp32s** \*pSrc, int nValue, **Npp32s** \*pDst, int nLength)  
*32-bit signed integer signal right shift with constant.*
- **NppStatus nppsRShiftC\_8u\_I** (int nValue, **Npp8u** \*pSrcDst, int nLength)  
*8-bit unsigned char in place signal right shift with constant.*
- **NppStatus nppsRShiftC\_16u\_I** (int nValue, **Npp16u** \*pSrcDst, int nLength)  
*16-bit unsigned short in place signal right shift with constant.*
- **NppStatus nppsRShiftC\_16s\_I** (int nValue, **Npp16s** \*pSrcDst, int nLength)  
*16-bit signed short in place signal right shift with constant.*
- **NppStatus nppsRShiftC\_32u\_I** (int nValue, **Npp32u** \*pSrcDst, int nLength)  
*32-bit unsigned signed integer in place signal right shift with constant.*
- **NppStatus nppsRShiftC\_32s\_I** (int nValue, **Npp32s** \*pSrcDst, int nLength)  
*32-bit signed signed integer in place signal right shift with constant.*

### 7.172.1 Detailed Description

Right shifts the bits of each sample of a signal by a constant amount.

### 7.172.2 Function Documentation

#### 7.172.2.1 **NppStatus nppsRShiftC\_16s** (const **Npp16s** \*pSrc, int nValue, **Npp16s** \*pDst, int nLength)

16-bit signed short signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.172.2.2 NppStatus nppsRShiftC\_16s\_I (int *nValue*, Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.172.2.3 NppStatus nppsRShiftC\_16u (const Npp16u \* *pSrc*, int *nValue*, Npp16u \* *pDst*, int *nLength*)**

16-bit unsigned short signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*pDst* Destination Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.172.2.4 NppStatus nppsRShiftC\_16u\_I (int *nValue*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.  
*nValue* Constant value to be used to right shift each vector element  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.172.2.5 NppStatus nppsRShiftC\_32s (const Npp32s \* *pSrc*, int *nValue*, Npp32s \* *pDst*, int *nLength*)

32-bit signed integer signal right shift with constant.

##### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.172.2.6 NppStatus nppsRShiftC\_32s\_I (int *nValue*, Npp32s \* *pSrcDst*, int *nLength*)

32-bit signed integer in place signal right shift with constant.

##### Parameters:

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.172.2.7 NppStatus nppsRShiftC\_32u (const Npp32u \* *pSrc*, int *nValue*, Npp32u \* *pDst*, int *nLength*)

32-bit unsigned integer signal right shift with constant.

##### Parameters:

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.172.2.8 NppStatus nppsRShiftC\_32u\_I (int *nValue*, Npp32u \* *pSrcDst*, int *nLength*)**

32-bit unsigned signed integer in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.172.2.9 NppStatus nppsRShiftC\_8u (const Npp8u \* *pSrc*, int *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char signal right shift with constant.

**Parameters:**

*pSrc* Source Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.172.2.10 NppStatus nppsRShiftC\_8u\_I (int *nValue*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit unsigned char in place signal right shift with constant.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nValue* Constant value to be used to right shift each vector element

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.173 Conversion Functions

### Modules

- [Convert](#)
- [Threshold](#)



## 7.174 Convert

### Convert

Routines for converting the sample-data type of signals.

- `NppStatus nppsConvert_8s16s` (const `Npp8s` \*pSrc, `Npp16s` \*pDst, int nLength)
- `NppStatus nppsConvert_8s32f` (const `Npp8s` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_8u32f` (const `Npp8u` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_16s8s_Sfs` (const `Npp16s` \*pSrc, `Npp8s` \*pDst, `Npp32u` nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_16s32s` (const `Npp16s` \*pSrc, `Npp32s` \*pDst, int nLength)
- `NppStatus nppsConvert_16s32f` (const `Npp16s` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_16u32f` (const `Npp16u` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_32s16s` (const `Npp32s` \*pSrc, `Npp16s` \*pDst, int nLength)
- `NppStatus nppsConvert_32s32f` (const `Npp32s` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_32s64f` (const `Npp32s` \*pSrc, `Npp64f` \*pDst, int nLength)
- `NppStatus nppsConvert_32f64f` (const `Npp32f` \*pSrc, `Npp64f` \*pDst, int nLength)
- `NppStatus nppsConvert_64s64f` (const `Npp64s` \*pSrc, `Npp64f` \*pDst, int nLength)
- `NppStatus nppsConvert_64f32f` (const `Npp64f` \*pSrc, `Npp32f` \*pDst, int nLength)
- `NppStatus nppsConvert_16s32f_Sfs` (const `Npp16s` \*pSrc, `Npp32f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_16s64f_Sfs` (const `Npp16s` \*pSrc, `Npp64f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s16s_Sfs` (const `Npp32s` \*pSrc, `Npp16s` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s32f_Sfs` (const `Npp32s` \*pSrc, `Npp32f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32s64f_Sfs` (const `Npp32s` \*pSrc, `Npp64f` \*pDst, int nLength, int nScaleFactor)
- `NppStatus nppsConvert_32f8s_Sfs` (const `Npp32f` \*pSrc, `Npp8s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f8u_Sfs` (const `Npp32f` \*pSrc, `Npp8u` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f16s_Sfs` (const `Npp32f` \*pSrc, `Npp16s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f16u_Sfs` (const `Npp32f` \*pSrc, `Npp16u` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_32f32s_Sfs` (const `Npp32f` \*pSrc, `Npp32s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64s32s_Sfs` (const `Npp64s` \*pSrc, `Npp32s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f16s_Sfs` (const `Npp64f` \*pSrc, `Npp16s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f32s_Sfs` (const `Npp64f` \*pSrc, `Npp32s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)
- `NppStatus nppsConvert_64f64s_Sfs` (const `Npp64f` \*pSrc, `Npp64s` \*pDst, int nLength, `NppRoundMode` eRoundMode, int nScaleFactor)



**7.174.1 Function Documentation**

- 7.174.1.1 **NppStatus** **nppsConvert\_16s32f** (const Npp16s \* *pSrc*, Npp32f \* *pDst*, int *nLength*)
- 7.174.1.2 **NppStatus** **nppsConvert\_16s32f\_Sfs** (const Npp16s \* *pSrc*, Npp32f \* *pDst*, int *nLength*, int *nScaleFactor*)
- 7.174.1.3 **NppStatus** **nppsConvert\_16s32s** (const Npp16s \* *pSrc*, Npp32s \* *pDst*, int *nLength*)
- 7.174.1.4 **NppStatus** **nppsConvert\_16s64f\_Sfs** (const Npp16s \* *pSrc*, Npp64f \* *pDst*, int *nLength*, int *nScaleFactor*)
- 7.174.1.5 **NppStatus** **nppsConvert\_16s8s\_Sfs** (const Npp16s \* *pSrc*, Npp8s \* *pDst*, Npp32u *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.6 **NppStatus** **nppsConvert\_16u32f** (const Npp16u \* *pSrc*, Npp32f \* *pDst*, int *nLength*)
- 7.174.1.7 **NppStatus** **nppsConvert\_32f16s\_Sfs** (const Npp32f \* *pSrc*, Npp16s \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.8 **NppStatus** **nppsConvert\_32f16u\_Sfs** (const Npp32f \* *pSrc*, Npp16u \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.9 **NppStatus** **nppsConvert\_32f32s\_Sfs** (const Npp32f \* *pSrc*, Npp32s \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.10 **NppStatus** **nppsConvert\_32f64f** (const Npp32f \* *pSrc*, Npp64f \* *pDst*, int *nLength*)
- 7.174.1.11 **NppStatus** **nppsConvert\_32f8s\_Sfs** (const Npp32f \* *pSrc*, Npp8s \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.12 **NppStatus** **nppsConvert\_32f8u\_Sfs** (const Npp32f \* *pSrc*, Npp8u \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.13 **NppStatus** **nppsConvert\_32s16s** (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*)
- 7.174.1.14 **NppStatus** **nppsConvert\_32s16s\_Sfs** (const Npp32s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, int *nScaleFactor*)
- 7.174.1.15 **NppStatus** **nppsConvert\_32s32f** (const Npp32s \* *pSrc*, Npp32f \* *pDst*, int *nLength*)
- 7.174.1.16 **NppStatus** **nppsConvert\_32s32f\_Sfs** (const Npp32s \* *pSrc*, Npp32f \* *pDst*, int *nLength*, int *nScaleFactor*)
- 7.174.1.17 **NppStatus** **nppsConvert\_32s64f** (const Npp32s \* *pSrc*, Npp64f \* *pDst*, int *nLength*)
- 7.174.1.18 **NppStatus** **nppsConvert\_32s64f\_Sfs** (const Npp32s \* *pSrc*, Npp64f \* *pDst*, int *nLength*, int *nScaleFactor*)
- 7.174.1.19 **NppStatus** **nppsConvert\_64f16s\_Sfs** (const Npp64f \* *pSrc*, Npp16s \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.20 **NppStatus** **nppsConvert\_64f32f** (const Npp64f \* *pSrc*, Npp32f \* *pDst*, int *nLength*)
- 7.174.1.21 **NppStatus** **nppsConvert\_64f32s\_Sfs** (const Npp64f \* *pSrc*, Npp32s \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.22 **NppStatus** **nppsConvert\_64f64s\_Sfs** (const Npp64f \* *pSrc*, Npp64s \* *pDst*, int *nLength*, NppRoundMode *eRoundMode*, int *nScaleFactor*)
- 7.174.1.23 **NppStatus** **nppsConvert\_64s32s\_Sfs** (const Npp64s \* *pSrc*, Npp32s \* *pDst*, int *nLength*,

## 7.175 Threshold

### Threshold Functions

Performs the threshold operation on the samples of a signal by limiting the sample values by a specified constant value.

- [NppStatus nppsThreshold\\_16s](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) \*pDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit signed short signal threshold with constant level.*
- [NppStatus nppsThreshold\\_16s\\_I](#) ([Npp16s](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit in place signed short signal threshold with constant level.*
- [NppStatus nppsThreshold\\_16sc](#) (const [Npp16sc](#) \*pSrc, [Npp16sc](#) \*pDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit signed short complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_16sc\\_I](#) ([Npp16sc](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel, [NppCmpOp](#) nRelOp)  
*16-bit in place signed short complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32f](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32f\\_I](#) ([Npp32f](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit in place floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32fc](#) (const [Npp32fc](#) \*pSrc, [Npp32fc](#) \*pDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit floating point complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_32fc\\_I](#) ([Npp32fc](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel, [NppCmpOp](#) nRelOp)  
*32-bit in place floating point complex number signal threshold with constant level.*
- [NppStatus nppsThreshold\\_64f](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) \*pDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)  
*64-bit floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_64f\\_I](#) ([Npp64f](#) \*pSrcDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)  
*64-bit in place floating point signal threshold with constant level.*
- [NppStatus nppsThreshold\\_64fc](#) (const [Npp64fc](#) \*pSrc, [Npp64fc](#) \*pDst, int nLength, [Npp64f](#) nLevel, [NppCmpOp](#) nRelOp)  
*64-bit floating point complex number signal threshold with constant level.*

- **NppStatus nppsThreshold\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel, **NppCmpOp** nRelOp)  
*64-bit in place floating point complex number signal threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_16s** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** nLevel)  
*16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_16s\_I** (**Npp16s** \*pSrcDst, int nLength, **Npp16s** nLevel)  
*16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_16sc** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, **Npp16s** nLevel)  
*16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_16sc\_I** (**Npp16sc** \*pSrcDst, int nLength, **Npp16s** nLevel)  
*16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength, **Npp32f** nLevel)  
*32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_32f\_I** (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nLevel)  
*32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength, **Npp32f** nLevel)  
*32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength, **Npp32f** nLevel)  
*32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength, **Npp64f** nLevel)  
*64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_64f\_I** (**Npp64f** \*pSrcDst, int nLength, **Npp64f** nLevel)  
*64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength, **Npp64f** nLevel)  
*64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LT\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel)  
*64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_GT\_16s** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** nLevel)  
*16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.*

- [NppStatus nppsThreshold\\_GT\\_16s\\_I](#) ([Npp16s](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel)  
*16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_16sc](#) (const [Npp16sc](#) \*pSrc, [Npp16sc](#) \*pDst, int nLength, [Npp16s](#) nLevel)  
*16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_16sc\\_I](#) ([Npp16sc](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel)  
*16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32f](#) (const [Npp32f](#) \*pSrc, [Npp32f](#) \*pDst, int nLength, [Npp32f](#) nLevel)  
*32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32f\\_I](#) ([Npp32f](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel)  
*32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32fc](#) (const [Npp32fc](#) \*pSrc, [Npp32fc](#) \*pDst, int nLength, [Npp32f](#) nLevel)  
*32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_32fc\\_I](#) ([Npp32fc](#) \*pSrcDst, int nLength, [Npp32f](#) nLevel)  
*32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64f](#) (const [Npp64f](#) \*pSrc, [Npp64f](#) \*pDst, int nLength, [Npp64f](#) nLevel)  
*64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64f\\_I](#) ([Npp64f](#) \*pSrcDst, int nLength, [Npp64f](#) nLevel)  
*64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64fc](#) (const [Npp64fc](#) \*pSrc, [Npp64fc](#) \*pDst, int nLength, [Npp64f](#) nLevel)  
*64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_GT\\_64fc\\_I](#) ([Npp64fc](#) \*pSrcDst, int nLength, [Npp64f](#) nLevel)  
*64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- [NppStatus nppsThreshold\\_LTV\\_16s](#) (const [Npp16s](#) \*pSrc, [Npp16s](#) \*pDst, int nLength, [Npp16s](#) nLevel, [Npp16s](#) nValue)  
*16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.*
- [NppStatus nppsThreshold\\_LTV\\_16s\\_I](#) ([Npp16s](#) \*pSrcDst, int nLength, [Npp16s](#) nLevel, [Npp16s](#) nValue)  
*16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.*
- [NppStatus nppsThreshold\\_LTV\\_16sc](#) (const [Npp16sc](#) \*pSrc, [Npp16sc](#) \*pDst, int nLength, [Npp16s](#) nLevel, [Npp16sc](#) nValue)  
*16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*

- **NppStatus nppsThreshold\_LTVal\_16sc\_I** (**Npp16sc** \*pSrcDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)  
*16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_32f\_I** (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_64f\_I** (**Npp64f** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_LTVal\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.*
- **NppStatus nppsThreshold\_GTVal\_16s** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** nLevel, **Npp16s** nValue)  
*16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVal\_16s\_I** (**Npp16s** \*pSrcDst, int nLength, **Npp16s** nLevel, **Npp16s** nValue)  
*16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVal\_16sc** (const **Npp16sc** \*pSrc, **Npp16sc** \*pDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)  
*16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVal\_16sc\_I** (**Npp16sc** \*pSrcDst, int nLength, **Npp16s** nLevel, **Npp16sc** nValue)  
*16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.*

- **NppStatus nppsThreshold\_GTVVal\_32f** (const **Npp32f** \*pSrc, **Npp32f** \*pDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_32f\_I** (**Npp32f** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32f** nValue)  
*32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_32fc** (const **Npp32fc** \*pSrc, **Npp32fc** \*pDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_32fc\_I** (**Npp32fc** \*pSrcDst, int nLength, **Npp32f** nLevel, **Npp32fc** nValue)  
*32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64f** (const **Npp64f** \*pSrc, **Npp64f** \*pDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64f\_I** (**Npp64f** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64f** nValue)  
*64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64fc** (const **Npp64fc** \*pSrc, **Npp64fc** \*pDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*
- **NppStatus nppsThreshold\_GTVVal\_64fc\_I** (**Npp64fc** \*pSrcDst, int nLength, **Npp64f** nLevel, **Npp64fc** nValue)  
*64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.*

### 7.175.1 Function Documentation

#### 7.175.1.1 **NppStatus nppsThreshold\_16s** (const **Npp16s** \*pSrc, **Npp16s** \*pDst, int nLength, **Npp16s** nLevel, **NppCmpOp** nRelOp)

16-bit signed short signal threshold with constant level.

##### Parameters:

**pSrc** Source Signal Pointer.

**pDst** Destination Signal Pointer.

**nLength** Signal Length.

**nLevel** Constant threshold value to be used to limit each signal sample

**nRelOp** NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).



**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.2 NppStatus nppsThreshold\_16s\_I (Npp16s \* *pSrcDst*, int *nLength*, Npp16s *nLevel*, NppCmpOp *nRelOp*)

16-bit in place signed short signal threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.3 NppStatus nppsThreshold\_16sc (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, Npp16s *nLevel*, NppCmpOp *nRelOp*)

16-bit signed short complex number signal threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.4 NppStatus nppsThreshold\_16sc\_I (Npp16sc \* *pSrcDst*, int *nLength*, Npp16s *nLevel*, NppCmpOp *nRelOp*)

16-bit in place signed short complex number signal threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

***nLevel*** Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

***nRelOp*** NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.5 NppStatus nppsThreshold\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)**

32-bit floating point signal threshold with constant level.

**Parameters:**

***pSrc*** [Source Signal Pointer](#).

***pDst*** [Destination Signal Pointer](#).

***nLength*** [Signal Length](#).

***nLevel*** Constant threshold value to be used to limit each signal sample

***nRelOp*** NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.6 NppStatus nppsThreshold\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)**

32-bit in place floating point signal threshold with constant level.

**Parameters:**

***pSrcDst*** [In-Place Signal Pointer](#).

***nLength*** [Signal Length](#).

***nLevel*** Constant threshold value to be used to limit each signal sample

***nRelOp*** NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.7 NppStatus nppsThreshold\_32fc (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)**

32-bit floating point complex number signal threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.8 NppStatus nppsThreshold\_32fc\_I (Npp32fc \* *pSrcDst*, int *nLength*, Npp32f *nLevel*, NppCmpOp *nRelOp*)

32-bit in place floating point complex number signal threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.9 NppStatus nppsThreshold\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)

64-bit floating point signal threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.10 **NppStatus nppsThreshold\_64f\_I** (Npp64f \* *pSrcDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)

64-bit in place floating point signal threshold with constant level.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.11 **NppStatus nppsThreshold\_64fc** (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)

64-bit floating point complex number signal threshold with constant level.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.12 **NppStatus nppsThreshold\_64fc\_I** (Npp64fc \* *pSrcDst*, int *nLength*, Npp64f *nLevel*, NppCmpOp *nRelOp*)

64-bit in place floating point complex number signal threshold with constant level.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nRelOp* NppCmpOp type of thresholding operation (NPP\_CMP\_LESS or NPP\_CMP\_GREATER only).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.13 NppStatus nppsThreshold\_GT\_16s (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, Npp16s *nLevel*)**

16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.14 NppStatus nppsThreshold\_GT\_16s\_I (Npp16s \* *pSrcDst*, int *nLength*, Npp16s *nLevel*)**

16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.15 NppStatus nppsThreshold\_GT\_16sc (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, Npp16s *nLevel*)**

16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.16 NppStatus nppsThreshold\_GT\_16sc\_I (Npp16sc \* *pSrcDst*, int *nLength*, Npp16s *nLevel*)**

16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.17 NppStatus nppsThreshold\_GT\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*, Npp32f *nLevel*)**

32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.18 NppStatus nppsThreshold\_GT\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*, Npp32f *nLevel*)**

32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.19 NppStatus nppsThreshold\_GT\_32fc (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*, Npp32f *nLevel*)**

32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.20 NppStatus nppsThreshold\_GT\_32fc\_I (Npp32fc \* *pSrcDst*, int *nLength*, Npp32f *nLevel*)**

32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.21 NppStatus nppsThreshold\_GT\_64f (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*, Npp64f *nLevel*)**

64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.22 NppStatus nppsThreshold\_GT\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*, Npp64f *nLevel*)**

64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.23 NppStatus nppsThreshold\_GT\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*, Npp64f *nLevel*)**

64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.24 NppStatus nppsThreshold\_GT\_64fc\_I (Npp64fc \* *pSrcDst*, int *nLength*, Npp64f *nLevel*)**

64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



**7.175.1.25 NppStatus nppsThreshold\_GTVal\_16s (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)**

16-bit signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.26 NppStatus nppsThreshold\_GTVal\_16s\_I (Npp16s \* *pSrcDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)**

16-bit in place signed short signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.27 NppStatus nppsThreshold\_GTVal\_16sc (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, Npp16s *nLevel*, Npp16sc *nValue*)**

16-bit signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.28 **NppStatus nppsThreshold\_GTVal\_16sc\_I** (Npp16sc \* *pSrcDst*, int *nLength*, Npp16s *nLevel*, Npp16sc *nValue*)

16-bit in place signed short complex number signal NPP\_CMP\_GREATER threshold with constant level.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.29 **NppStatus nppsThreshold\_GTVal\_32f** (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*, Npp32f *nLevel*, Npp32f *nValue*)

32-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.30 **NppStatus nppsThreshold\_GTVal\_32f\_I** (Npp32f \* *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32f *nValue*)

32-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.31 **NppStatus nppsThreshold\_GTVal\_32fc** (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.32 **NppStatus nppsThreshold\_GTVal\_32fc\_I** (Npp32fc \* *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

#### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.175.1.33 **NppStatus nppsThreshold\_GTVal\_64f** (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*, Npp64f *nLevel*, Npp64f *nValue*)

64-bit floating point signal NPP\_CMP\_GREATER threshold with constant level.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.34 **NppStatus nppsThreshold\_GTVal\_64f\_I (Npp64f \* pSrcDst, int nLength, Npp64f nLevel, Npp64f nValue)**

64-bit in place floating point signal NPP\_CMP\_GREATER threshold with constant level.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.35 **NppStatus nppsThreshold\_GTVal\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.36 **NppStatus nppsThreshold\_GTVal\_64fc\_I (Npp64fc \* pSrcDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit in place floating point complex number signal NPP\_CMP\_GREATER threshold with constant level.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.37 NppStatus nppsThreshold\_LT\_16s (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, Npp16s *nLevel*)**

16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.38 NppStatus nppsThreshold\_LT\_16s\_I (Npp16s \* *pSrcDst*, int *nLength*, Npp16s *nLevel*)**

16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.39 NppStatus nppsThreshold\_LT\_16sc (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, Npp16s *nLevel*)**

16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.175.1.40 **NppStatus nppsThreshold\_LT\_16sc\_I** (Npp16sc \* *pSrcDst*, int *nLength*, Npp16s *nLevel*)

16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.41 **NppStatus nppsThreshold\_LT\_32f** (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*, Npp32f *nLevel*)

32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.42 **NppStatus nppsThreshold\_LT\_32f\_I** (Npp32f \* *pSrcDst*, int *nLength*, Npp32f *nLevel*)

32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

##### Parameters:

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.175.1.43 **NppStatus nppsThreshold\_LT\_32fc** (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*, Npp32f *nLevel*)

32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

##### Parameters:

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.175.1.44 **NppStatus nppsThreshold\_LT\_32fc\_I** (Npp32fc \* *pSrcDst*, int *nLength*, Npp32f *nLevel*)

32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

##### Parameters:

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

#### 7.175.1.45 **NppStatus nppsThreshold\_LT\_64f** (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*, Npp64f *nLevel*)

64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

##### Parameters:

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

##### Returns:

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.46 NppStatus nppsThreshold\_LT\_64f\_I (Npp64f \* *pSrcDst*, int *nLength*, Npp64f *nLevel*)**

64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.47 NppStatus nppsThreshold\_LT\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*, Npp64f *nLevel*)**

64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.175.1.48 NppStatus nppsThreshold\_LT\_64fc\_I (Npp64fc \* *pSrcDst*, int *nLength*, Npp64f *nLevel*)**

64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



**7.175.1.49 NppStatus nppsThreshold\_LTVal\_16s (const Npp16s \* *pSrc*, Npp16s \* *pDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)**

16-bit signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.50 NppStatus nppsThreshold\_LTVal\_16s\_I (Npp16s \* *pSrcDst*, int *nLength*, Npp16s *nLevel*, Npp16s *nValue*)**

16-bit in place signed short signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.51 NppStatus nppsThreshold\_LTVal\_16sc (const Npp16sc \* *pSrc*, Npp16sc \* *pDst*, int *nLength*, Npp16s *nLevel*, Npp16sc *nValue*)**

16-bit signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.52 NppStatus nppsThreshold\_LTVVal\_16sc\_I (Npp16sc \* *pSrcDst*, int *nLength*, Npp16s *nLevel*, Npp16sc *nValue*)**

16-bit in place signed short complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.53 NppStatus nppsThreshold\_LTVVal\_32f (const Npp32f \* *pSrc*, Npp32f \* *pDst*, int *nLength*, Npp32f *nLevel*, Npp32f *nValue*)**

32-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.54 NppStatus nppsThreshold\_LTVVal\_32f\_I (Npp32f \* *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32f *nValue*)**

32-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.55** **NppStatus nppsThreshold\_LTVal\_32fc** (const Npp32fc \* *pSrc*, Npp32fc \* *pDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.56** **NppStatus nppsThreshold\_LTVal\_32fc\_I** (Npp32fc \* *pSrcDst*, int *nLength*, Npp32f *nLevel*, Npp32fc *nValue*)

32-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.57** **NppStatus nppsThreshold\_LTVal\_64f** (const Npp64f \* *pSrc*, Npp64f \* *pDst*, int *nLength*, Npp64f *nLevel*, Npp64f *nValue*)

64-bit floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.58 NppStatus nppsThreshold\_LTVal\_64f\_I (Npp64f \* pSrcDst, int nLength, Npp64f nLevel, Npp64f nValue)**

64-bit in place floating point signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.59 NppStatus nppsThreshold\_LTVal\_64fc (const Npp64fc \* pSrc, Npp64fc \* pDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.175.1.60 NppStatus nppsThreshold\_LTVal\_64fc\_I (Npp64fc \* pSrcDst, int nLength, Npp64f nLevel, Npp64fc nValue)**

64-bit in place floating point complex number signal NPP\_CMP\_LESS threshold with constant level.

**Parameters:**

*pSrcDst* [In-Place Signal Pointer](#).

*nLength* [Signal Length](#).

*nLevel* Constant threshold value (real part only and must be greater than 0) to be used to limit each signal sample

*nValue* Constant value to replace source value when threshold test is true.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.176 Filtering Functions

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

### Modules

- [Integral](#)

*Compute the indefinite interal of a given signal.*

### 7.176.1 Detailed Description

Functions that provide functionality of generating output signal based on the input signal like signal integral, etc.

## 7.177 Integral

Compute the indefinite integral of a given signal.

### Functions

- `NppStatus nppsIntegralGetBufferSize_32s` (int *nLength*, int \**hpBufferSize*)
- `NppStatus nppsIntegral_32s` (const `Npp32s` \**pSrc*, `Npp32s` \**pDst*, int *nLength*, `Npp8u` \**pDeviceBuffer*)

### 7.177.1 Detailed Description

Compute the indefinite integral of a given signal.

The i-th element is computed to be

$$s'_i = \sum_0^i s_j$$

### 7.177.2 Function Documentation

**7.177.2.1** `NppStatus nppsIntegral_32s` (const `Npp32s` \**pSrc*, `Npp32s` \**pDst*, int *nLength*, `Npp8u` \**pDeviceBuffer*)

**7.177.2.2** `NppStatus nppsIntegralGetBufferSize_32s` (int *nLength*, int \**hpBufferSize*)

## 7.178 Initialization

### Modules

- [Set](#)
- [Zero](#)
- [Copy](#)

## 7.179 Set

### Set

Set methods for 1D vectors of various types.

The copy methods operate on vector data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to Npp8u type) and length of the vectors, i.e. the number of items.

- [NppStatus nppsSet\\_8u](#) ([Npp8u](#) nValue, [Npp8u](#) \*pDst, int nLength)  
*8-bit unsigned char, vector set method.*
- [NppStatus nppsSet\\_8s](#) ([Npp8s](#) nValue, [Npp8s](#) \*pDst, int nLength)  
*8-bit signed char, vector set method.*
- [NppStatus nppsSet\\_16u](#) ([Npp16u](#) nValue, [Npp16u](#) \*pDst, int nLength)  
*16-bit unsigned integer, vector set method.*
- [NppStatus nppsSet\\_16s](#) ([Npp16s](#) nValue, [Npp16s](#) \*pDst, int nLength)  
*16-bit signed integer, vector set method.*
- [NppStatus nppsSet\\_16sc](#) ([Npp16sc](#) nValue, [Npp16sc](#) \*pDst, int nLength)  
*16-bit integer complex, vector set method.*
- [NppStatus nppsSet\\_32u](#) ([Npp32u](#) nValue, [Npp32u](#) \*pDst, int nLength)  
*32-bit unsigned integer, vector set method.*
- [NppStatus nppsSet\\_32s](#) ([Npp32s](#) nValue, [Npp32s](#) \*pDst, int nLength)  
*32-bit signed integer, vector set method.*
- [NppStatus nppsSet\\_32sc](#) ([Npp32sc](#) nValue, [Npp32sc](#) \*pDst, int nLength)  
*32-bit integer complex, vector set method.*
- [NppStatus nppsSet\\_32f](#) ([Npp32f](#) nValue, [Npp32f](#) \*pDst, int nLength)  
*32-bit float, vector set method.*
- [NppStatus nppsSet\\_32fc](#) ([Npp32fc](#) nValue, [Npp32fc](#) \*pDst, int nLength)  
*32-bit float complex, vector set method.*
- [NppStatus nppsSet\\_64s](#) ([Npp64s](#) nValue, [Npp64s](#) \*pDst, int nLength)  
*64-bit long long integer, vector set method.*
- [NppStatus nppsSet\\_64sc](#) ([Npp64sc](#) nValue, [Npp64sc](#) \*pDst, int nLength)  
*64-bit long long integer complex, vector set method.*
- [NppStatus nppsSet\\_64f](#) ([Npp64f](#) nValue, [Npp64f](#) \*pDst, int nLength)  
*64-bit double, vector set method.*
- [NppStatus nppsSet\\_64fc](#) ([Npp64fc](#) nValue, [Npp64fc](#) \*pDst, int nLength)  
*64-bit double complex, vector set method.*



## 7.179.1 Function Documentation

### 7.179.1.1 NppStatus nppsSet\_16s (Npp16s *nValue*, Npp16s \* *pDst*, int *nLength*)

16-bit signed integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.179.1.2 NppStatus nppsSet\_16sc (Npp16sc *nValue*, Npp16sc \* *pDst*, int *nLength*)

16-bit integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.179.1.3 NppStatus nppsSet\_16u (Npp16u *nValue*, Npp16u \* *pDst*, int *nLength*)

16-bit unsigned integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

### 7.179.1.4 NppStatus nppsSet\_32f (Npp32f *nValue*, Npp32f \* *pDst*, int *nLength*)

32-bit float, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.179.1.5 NppStatus nppsSet\_32fc (Npp32fc *nValue*, Npp32fc \* *pDst*, int *nLength*)**

32-bit float complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.179.1.6 NppStatus nppsSet\_32s (Npp32s *nValue*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.179.1.7 NppStatus nppsSet\_32sc (Npp32sc *nValue*, Npp32sc \* *pDst*, int *nLength*)**

32-bit integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.179.1.8 NppStatus nppsSet\_32u (Npp32u *nValue*, Npp32u \* *pDst*, int *nLength*)**

32-bit unsigned integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.179.1.9 NppStatus nppsSet\_64f (Npp64f *nValue*, Npp64f \* *pDst*, int *nLength*)**

64-bit double, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.179.1.10 NppStatus nppsSet\_64fc (Npp64fc *nValue*, Npp64fc \* *pDst*, int *nLength*)**

64-bit double complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.179.1.11 NppStatus nppsSet\_64s (Npp64s *nValue*, Npp64s \* *pDst*, int *nLength*)**

64-bit long long integer, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* Destination Signal Pointer.

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.179.1.12 NppStatus nppsSet\_64sc (Npp64sc *nValue*, Npp64sc \* *pDst*, int *nLength*)**

64-bit long long integer complex, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.179.1.13 NppStatus nppsSet\_8s (Npp8s *nValue*, Npp8s \* *pDst*, int *nLength*)**

8-bit signed char, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.179.1.14 NppStatus nppsSet\_8u (Npp8u *nValue*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char, vector set method.

**Parameters:**

*nValue* Value used to initialize the vector pDst.

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

## 7.180 Zero

### Zero

Set signals to zero.

- [NppStatus nppsZero\\_8u](#) ([Npp8u](#) \*pDst, int nLength)  
*8-bit unsigned char, vector zero method.*
- [NppStatus nppsZero\\_16s](#) ([Npp16s](#) \*pDst, int nLength)  
*16-bit integer, vector zero method.*
- [NppStatus nppsZero\\_16sc](#) ([Npp16sc](#) \*pDst, int nLength)  
*16-bit integer complex, vector zero method.*
- [NppStatus nppsZero\\_32s](#) ([Npp32s](#) \*pDst, int nLength)  
*32-bit integer, vector zero method.*
- [NppStatus nppsZero\\_32sc](#) ([Npp32sc](#) \*pDst, int nLength)  
*32-bit integer complex, vector zero method.*
- [NppStatus nppsZero\\_32f](#) ([Npp32f](#) \*pDst, int nLength)  
*32-bit float, vector zero method.*
- [NppStatus nppsZero\\_32fc](#) ([Npp32fc](#) \*pDst, int nLength)  
*32-bit float complex, vector zero method.*
- [NppStatus nppsZero\\_64s](#) ([Npp64s](#) \*pDst, int nLength)  
*64-bit long long integer, vector zero method.*
- [NppStatus nppsZero\\_64sc](#) ([Npp64sc](#) \*pDst, int nLength)  
*64-bit long long integer complex, vector zero method.*
- [NppStatus nppsZero\\_64f](#) ([Npp64f](#) \*pDst, int nLength)  
*64-bit double, vector zero method.*
- [NppStatus nppsZero\\_64fc](#) ([Npp64fc](#) \*pDst, int nLength)  
*64-bit double complex, vector zero method.*

### 7.180.1 Function Documentation

#### 7.180.1.1 NppStatus nppsZero\_16s (Npp16s \* pDst, int nLength)

16-bit integer, vector zero method.

##### Parameters:

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.180.1.2 NppStatus nppsZero\_16sc (Npp16sc \* *pDst*, int *nLength*)**

16-bit integer complex, vector zero method.

**Parameters:**

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.180.1.3 NppStatus nppsZero\_32f (Npp32f \* *pDst*, int *nLength*)**

32-bit float, vector zero method.

**Parameters:**

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.180.1.4 NppStatus nppsZero\_32fc (Npp32fc \* *pDst*, int *nLength*)**

32-bit float complex, vector zero method.

**Parameters:**

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.180.1.5 NppStatus nppsZero\_32s (Npp32s \* *pDst*, int *nLength*)**

32-bit integer, vector zero method.

**Parameters:**

*pDst* [Destination Signal Pointer](#).

*nLength* [Signal Length](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.180.1.6 NppStatus nppsZero\_32sc (Npp32sc \* *pDst*, int *nLength*)**

32-bit integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.180.1.7 NppStatus nppsZero\_64f (Npp64f \* *pDst*, int *nLength*)**

64-bit double, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.180.1.8 NppStatus nppsZero\_64fc (Npp64fc \* *pDst*, int *nLength*)**

64-bit double complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.180.1.9 NppStatus nppsZero\_64s (Npp64s \* *pDst*, int *nLength*)**

64-bit long long integer, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.180.1.10 NppStatus nppsZero\_64sc (Npp64sc \* *pDst*, int *nLength*)**

64-bit long long integer complex, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.180.1.11 NppStatus nppsZero\_8u (Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char, vector zero method.

**Parameters:**

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.



## 7.181 Copy

### Copy

Copy methods for various type signals.

Copy methods operate on signal data given as a pointer to the underlying data-type (e.g. 8-bit vectors would be passed as pointers to `Npp8u` type) and length of the vectors, i.e. the number of items.

- `NppStatus nppsCopy_8u` (const `Npp8u` \*pSrc, `Npp8u` \*pDst, int nLength)  
*8-bit unsigned char, vector copy method*
- `NppStatus nppsCopy_16s` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength)  
*16-bit signed short, vector copy method.*
- `NppStatus nppsCopy_32s` (const `Npp32s` \*pSrc, `Npp32s` \*pDst, int nLength)  
*32-bit signed integer, vector copy method.*
- `NppStatus nppsCopy_32f` (const `Npp32f` \*pSrc, `Npp32f` \*pDst, int nLength)  
*32-bit float, vector copy method.*
- `NppStatus nppsCopy_64s` (const `Npp64s` \*pSrc, `Npp64s` \*pDst, int nLength)  
*64-bit signed integer, vector copy method.*
- `NppStatus nppsCopy_16sc` (const `Npp16sc` \*pSrc, `Npp16sc` \*pDst, int nLength)  
*16-bit complex short, vector copy method.*
- `NppStatus nppsCopy_32sc` (const `Npp32sc` \*pSrc, `Npp32sc` \*pDst, int nLength)  
*32-bit complex signed integer, vector copy method.*
- `NppStatus nppsCopy_32fc` (const `Npp32fc` \*pSrc, `Npp32fc` \*pDst, int nLength)  
*32-bit complex float, vector copy method.*
- `NppStatus nppsCopy_64sc` (const `Npp64sc` \*pSrc, `Npp64sc` \*pDst, int nLength)  
*64-bit complex signed integer, vector copy method.*
- `NppStatus nppsCopy_64fc` (const `Npp64fc` \*pSrc, `Npp64fc` \*pDst, int nLength)  
*64-bit complex double, vector copy method.*

### 7.181.1 Function Documentation

#### 7.181.1.1 `NppStatus nppsCopy_16s` (const `Npp16s` \*pSrc, `Npp16s` \*pDst, int nLength)

16-bit signed short, vector copy method.

#### Parameters:

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.2 NppStatus nppsCopy\_16sc (const Npp16sc \* pSrc, Npp16sc \* pDst, int nLength)**

16-bit complex short, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.3 NppStatus nppsCopy\_32f (const Npp32f \* pSrc, Npp32f \* pDst, int nLength)**

32-bit float, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.4 NppStatus nppsCopy\_32fc (const Npp32fc \* pSrc, Npp32fc \* pDst, int nLength)**

32-bit complex float, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.5 NppStatus nppsCopy\_32s (const Npp32s \* *pSrc*, Npp32s \* *pDst*, int *nLength*)**

32-bit signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.6 NppStatus nppsCopy\_32sc (const Npp32sc \* *pSrc*, Npp32sc \* *pDst*, int *nLength*)**

32-bit complex signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.7 NppStatus nppsCopy\_64fc (const Npp64fc \* *pSrc*, Npp64fc \* *pDst*, int *nLength*)**

64-bit complex double, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.8 NppStatus nppsCopy\_64s (const Npp64s \* *pSrc*, Npp64s \* *pDst*, int *nLength*)**

64-bit signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.9 NppStatus nppsCopy\_64sc (const Npp64sc \* *pSrc*, Npp64sc \* *pDst*, int *nLength*)**

64-bit complex signed integer, vector copy method.

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.181.1.10 NppStatus nppsCopy\_8u (const Npp8u \* *pSrc*, Npp8u \* *pDst*, int *nLength*)**

8-bit unsigned char, vector copy method

**Parameters:**

*pSrc* Source Signal Pointer.

*pDst* Destination Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.182 Statistical Functions

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

### Modules

- [MinEvery And MaxEvery Functions](#)

*Performs the min or max operation on the samples of a signal.*

- [Sum](#)

*signal\_min\_every\_or\_max\_every*

- [Maximum](#)
- [Minimum](#)
- [Mean](#)
- [Standard Deviation](#)
- [Mean And Standard Deviation](#)
- [Minimum\\_Maximum](#)
- [Infinity Norm](#)
- [L1 Norm](#)
- [L2 Norm](#)
- [Infinity Norm Diff](#)
- [L1 Norm Diff](#)
- [L2 Norm Diff](#)
- [Dot Product](#)
- [Count In Range](#)
- [Count Zero Crossings](#)
- [MaximumError](#)

*Primitives for computing the maximum error between two signals.*

- [AverageError](#)

*Primitives for computing the Average error between two signals.*

- [MaximumRelativeError](#)

*Primitives for computing the MaximumRelative error between two signals.*

- [AverageRelativeError](#)

*Primitives for computing the AverageRelative error between two signals.*

### 7.182.1 Detailed Description

Functions that provide global signal statistics like: sum, mean, standard deviation, min, max, etc.

## 7.183 MinEvery And MaxEvery Functions

Performs the min or max operation on the samples of a signal.

### Functions

- `NppStatus nppsMinEvery_8u_I` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength)  
*8-bit in place min value for each pair of elements.*
- `NppStatus nppsMinEvery_16u_I` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short integer in place min value for each pair of elements.*
- `NppStatus nppsMinEvery_16s_I` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength)  
*16-bit signed short integer in place min value for each pair of elements.*
- `NppStatus nppsMinEvery_32s_I` (const `Npp32s` \*pSrc, `Npp32s` \*pSrcDst, int nLength)  
*32-bit signed integer in place min value for each pair of elements.*
- `NppStatus nppsMinEvery_32f_I` (const `Npp32f` \*pSrc, `Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point in place min value for each pair of elements.*
- `NppStatus nppsMinEvery_64f_I` (const `Npp64f` \*pSrc, `Npp64f` \*pSrcDst, int nLength)  
*64-bit floating point in place min value for each pair of elements.*
- `NppStatus nppsMaxEvery_8u_I` (const `Npp8u` \*pSrc, `Npp8u` \*pSrcDst, int nLength)  
*8-bit in place max value for each pair of elements.*
- `NppStatus nppsMaxEvery_16u_I` (const `Npp16u` \*pSrc, `Npp16u` \*pSrcDst, int nLength)  
*16-bit unsigned short integer in place max value for each pair of elements.*
- `NppStatus nppsMaxEvery_16s_I` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength)  
*16-bit signed short integer in place max value for each pair of elements.*
- `NppStatus nppsMaxEvery_32s_I` (const `Npp32s` \*pSrc, `Npp32s` \*pSrcDst, int nLength)  
*32-bit signed integer in place max value for each pair of elements.*
- `NppStatus nppsMaxEvery_32f_I` (const `Npp32f` \*pSrc, `Npp32f` \*pSrcDst, int nLength)  
*32-bit floating point in place max value for each pair of elements.*

### 7.183.1 Detailed Description

Performs the min or max operation on the samples of a signal.

### 7.183.2 Function Documentation

#### 7.183.2.1 `NppStatus nppsMaxEvery_16s_I` (const `Npp16s` \*pSrc, `Npp16s` \*pSrcDst, int nLength)

16-bit signed short integer in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.2 NppStatus nppsMaxEvery\_16u\_I (const Npp16u \* pSrc, Npp16u \* pSrcDst, int nLength)**

16-bit unsigned short integer in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.3 NppStatus nppsMaxEvery\_32f\_I (const Npp32f \* pSrc, Npp32f \* pSrcDst, int nLength)**

32-bit floating point in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.4 NppStatus nppsMaxEvery\_32s\_I (const Npp32s \* pSrc, Npp32s \* pSrcDst, int nLength)**

32-bit signed integer in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.  
*pSrcDst* In-Place Signal Pointer.  
*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.5 NppStatus nppsMaxEvery\_8u\_I (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit in place max value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.6 NppStatus nppsMinEvery\_16s\_I (const Npp16s \* *pSrc*, Npp16s \* *pSrcDst*, int *nLength*)**

16-bit signed short integer in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.7 NppStatus nppsMinEvery\_16u\_I (const Npp16u \* *pSrc*, Npp16u \* *pSrcDst*, int *nLength*)**

16-bit unsigned short integer in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.8 NppStatus nppsMinEvery\_32f\_I (const Npp32f \* *pSrc*, Npp32f \* *pSrcDst*, int *nLength*)**

32-bit floating point in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.



*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.9 NppStatus nppsMinEvery\_32s\_I (const Npp32s \* *pSrc*, Npp32s \* *pSrcDst*, int *nLength*)**

32-bit signed integer in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.10 NppStatus nppsMinEvery\_64f\_I (const Npp64f \* *pSrc*, Npp64f \* *pSrcDst*, int *nLength*)**

64-bit floating point in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

**7.183.2.11 NppStatus nppsMinEvery\_8u\_I (const Npp8u \* *pSrc*, Npp8u \* *pSrcDst*, int *nLength*)**

8-bit in place min value for each pair of elements.

**Parameters:**

*pSrc* Source Signal Pointer.

*pSrcDst* In-Place Signal Pointer.

*nLength* Signal Length.

**Returns:**

Signal Data Related Error Codes, Length Related Error Codes.

## 7.184 Sum

signal\_min\_every\_or\_max\_every

### Functions

- [NppStatus nppsSumGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_32f.*
- [NppStatus nppsSumGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_32fc.*
- [NppStatus nppsSumGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_64f.*
- [NppStatus nppsSumGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_64fc.*
- [NppStatus nppsSumGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16s\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16sc\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_16sc32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16sc32sc\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_32s\_Sfs.*
- [NppStatus nppsSumGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsSum\_16s32s\_Sfs.*
- [NppStatus nppsSum\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector sum method*
- [NppStatus nppsSum\\_32fc](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp32fc](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector sum method*
- [NppStatus nppsSum\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double vector sum method*
- [NppStatus nppsSum\\_64fc](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64fc](#) \*pSum, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double complex vector sum method*

- **NppStatus nppsSum\_16s\_Sfs** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit short vector sum with integer scaling method*
- **NppStatus nppsSum\_32s\_Sfs** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*32-bit integer vector sum with integer scaling method*
- **NppStatus nppsSum\_16sc\_Sfs** (const **Npp16sc** \*pSrc, int nLength, **Npp16sc** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit short complex vector sum with integer scaling method*
- **NppStatus nppsSum\_16sc32sc\_Sfs** (const **Npp16sc** \*pSrc, int nLength, **Npp32sc** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit short complex vector sum (32bit int complex) with integer scaling method*
- **NppStatus nppsSum\_16s32s\_Sfs** (const **Npp16s** \*pSrc, int nLength, **Npp32s** \*pSum, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit integer vector sum (32bit) with integer scaling method*

### 7.184.1 Detailed Description

signal\_min\_every\_or\_max\_every

### 7.184.2 Function Documentation

#### 7.184.2.1 NppStatus nppsSum\_16s32s\_Sfs (const Npp16s \*pSrc, int nLength, Npp32s \*pSum, int nScaleFactor, Npp8u \*pDeviceBuffer)

16-bit integer vector sum (32bit) with integer scaling method

##### Parameters:

**pSrc** Source Signal Pointer.

**nLength** Signal Length.

**pSum** Pointer to the output result.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

**nScaleFactor** Integer Result Scaling.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.184.2.2 NppStatus nppsSum\_16s\_Sfs (const Npp16s \*pSrc, int nLength, Npp16s \*pSum, int nScaleFactor, Npp8u \*pDeviceBuffer)

16-bit short vector sum with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.184.2.3** `NppStatus nppsSum_16sc32sc_Sfs (const Npp16sc * pSrc, int nLength, Npp32sc * pSum, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector sum (32bit int complex) with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_16sc32sc\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.184.2.4** `NppStatus nppsSum_16sc_Sfs (const Npp16sc * pSrc, int nLength, Npp16sc * pSum, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit short complex vector sum with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_16sc\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.184.2.5 NppStatus nppsSum\_32f (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pSum*, Npp8u \* *pDeviceBuffer*)

32-bit float vector sum method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.184.2.6 NppStatus nppsSum\_32fc (const Npp32fc \* *pSrc*, int *nLength*, Npp32fc \* *pSum*, Npp8u \* *pDeviceBuffer*)

32-bit float complex vector sum method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.184.2.7 NppStatus nppsSum\_32s\_Sfs (const Npp32s \* *pSrc*, int *nLength*, Npp32s \* *pSum*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

32-bit integer vector sum with integer scaling method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_32s\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.184.2.8 NppStatus nppsSum\_64f (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pSum*, Npp8u \* *pDeviceBuffer*)

64-bit double vector sum method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_64f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.184.2.9 NppStatus nppsSum\_64fc (const Npp64fc \* *pSrc*, int *nLength*, Npp64fc \* *pSum*, Npp8u \* *pDeviceBuffer*)

64-bit double complex vector sum method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pSum* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsSumGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.184.2.10 NppStatus nppsSumGetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsSum\_16s32s\_Sfs.

##### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

##### Returns:

NPP\_SUCCESS

**7.184.2.11 NppStatus nppsSumGetBufferSize\_16s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.184.2.12 NppStatus nppsSumGetBufferSize\_16sc32sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_16sc32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.184.2.13 NppStatus nppsSumGetBufferSize\_16sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_16sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.184.2.14 NppStatus nppsSumGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.184.2.15 NppStatus nppsSumGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.184.2.16 NppStatus nppsSumGetBufferSize\_32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.184.2.17 NppStatus nppsSumGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.184.2.18 NppStatus nppsSumGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsSum\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS



## 7.185 Maximum

### Functions

- [NppStatus nppsMaxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_16s.*
- [NppStatus nppsMaxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_32s.*
- [NppStatus nppsMaxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_32f.*
- [NppStatus nppsMaxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMax\_64f.*
- [NppStatus nppsMax\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector max method*
- [NppStatus nppsMax\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector max method*
- [NppStatus nppsMax\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector max method*
- [NppStatus nppsMax\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector max method*
- [NppStatus nppsMaxIndxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_16s.*
- [NppStatus nppsMaxIndxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_32s.*
- [NppStatus nppsMaxIndxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_32f.*
- [NppStatus nppsMaxIndxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMaxIndx\_64f.*
- [NppStatus nppsMaxIndx\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMax, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector max index method*
- [NppStatus nppsMaxIndx\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMax, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*32-bit integer vector max index method*

- **NppStatus nppsMaxIndx\_32f** (const **Npp32f** \*pSrc, int nLength, **Npp32f** \*pMax, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*32-bit float vector max index method*

- **NppStatus nppsMaxIndx\_64f** (const **Npp64f** \*pSrc, int nLength, **Npp64f** \*pMax, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*64-bit float vector max index method*

- **NppStatus nppsMaxAbsGetBufferSize\_16s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbs\_16s.*

- **NppStatus nppsMaxAbsGetBufferSize\_32s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbs\_32s.*

- **NppStatus nppsMaxAbs\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMaxAbs, **Npp8u** \*pDeviceBuffer)

*16-bit integer vector max absolute method*

- **NppStatus nppsMaxAbs\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMaxAbs, **Npp8u** \*pDeviceBuffer)

*32-bit integer vector max absolute method*

- **NppStatus nppsMaxAbsIndxGetBufferSize\_16s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_16s.*

- **NppStatus nppsMaxAbsIndxGetBufferSize\_32s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_32s.*

- **NppStatus nppsMaxAbsIndx\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMaxAbs, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*16-bit integer vector max absolute index method*

- **NppStatus nppsMaxAbsIndx\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMaxAbs, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*32-bit integer vector max absolute index method*

## 7.185.1 Function Documentation

### 7.185.1.1 **NppStatus nppsMax\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMax, **Npp8u** \*pDeviceBuffer)

16-bit integer vector max method

#### Parameters:

**pSrc** Source Signal Pointer.

**nLength** Signal Length.

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.2 NppStatus nppsMax\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMax, Npp8u \* pDeviceBuffer)**

32-bit float vector max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.3 NppStatus nppsMax\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMax, Npp8u \* pDeviceBuffer)**

32-bit integer vector max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.4 NppStatus nppsMax\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMax, Npp8u \* pDeviceBuffer)**

64-bit float vector max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.5 NppStatus nppsMaxAbs\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMaxAbs, Npp8u \* pDeviceBuffer)**

16-bit integer vector max absolute method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMaxAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.6 NppStatus nppsMaxAbs\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMaxAbs, Npp8u \* pDeviceBuffer)**

32-bit integer vector max absolute method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMaxAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.7 NppStatus nppsMaxAbsGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMaxAbs\_16s.

**Parameters:**

*nLength* [Signal Length](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

### 7.185.1.8 NppStatus nppsMaxAbsGetBufferSize\_32s (int nLength, int \* hpBufferSize)

Device scratch buffer size (in bytes) for nppsMaxAbs\_32s.

**Parameters:**

**nLength** [Signal Length](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

### 7.185.1.9 NppStatus nppsMaxAbsIndx\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMaxAbs, int \* pIndx, Npp8u \* pDeviceBuffer)

16-bit integer vector max absolute index method

**Parameters:**

**pSrc** [Source Signal Pointer](#).

**nLength** [Signal Length](#).

**pMaxAbs** Pointer to the output result.

**pIndx** Pointer to the index value of the first maximum element.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsIndxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.185.1.10 NppStatus nppsMaxAbsIndx\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMaxAbs, int \* pIndx, Npp8u \* pDeviceBuffer)

32-bit integer vector max absolute index method

**Parameters:**

**pSrc** [Source Signal Pointer](#).

**nLength** [Signal Length](#).

**pMaxAbs** Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxAbsIndxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.11 NppStatus nppsMaxAbsIndxGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.185.1.12 NppStatus nppsMaxAbsIndxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMaxAbsIndx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.185.1.13 NppStatus nppsMaxGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMax\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.185.1.14 NppStatus nppsMaxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMax\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.185.1.15 NppStatus nppsMaxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMax\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.185.1.16 NppStatus nppsMaxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMax\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.185.1.17 NppStatus nppsMaxIndx\_16s (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMax*, int \* *pIndx*, Npp8u \* *pDeviceBuffer*)**

16-bit integer vector max index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.18** `NppStatus nppsMaxIndx_32f (const Npp32f * pSrc, int nLength, Npp32f * pMax, int * pIndx, Npp8u * pDeviceBuffer)`

32-bit float vector max index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.185.1.19** `NppStatus nppsMaxIndx_32s (const Npp32s * pSrc, int nLength, Npp32s * pMax, int * pIndx, Npp8u * pDeviceBuffer)`

32-bit integer vector max index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



### 7.185.1.20 **NppStatus nppsMaxIndx\_64f** (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pMax*, int \* *pIndx*, Npp8u \* *pDeviceBuffer*)

64-bit float vector max index method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMax* Pointer to the output result.

*pIndx* Pointer to the index value of the first maximum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaxIndxGetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.185.1.21 **NppStatus nppsMaxIndxGetBufferSize\_16s** (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxIndx\_16s.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_SUCCESS

### 7.185.1.22 **NppStatus nppsMaxIndxGetBufferSize\_32f** (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMaxIndx\_32f.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_SUCCESS

**7.185.1.23 NppStatus nppsMaxIndxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMaxIndx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.185.1.24 NppStatus nppsMaxIndxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMaxIndx\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.186 Minimum

### Functions

- [NppStatus nppsMinGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_16s.*
- [NppStatus nppsMinGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_32s.*
- [NppStatus nppsMinGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_32f.*
- [NppStatus nppsMinGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMin\_64f.*
- [NppStatus nppsMin\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector min method*
- [NppStatus nppsMin\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector min method*
- [NppStatus nppsMin\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector min method*
- [NppStatus nppsMin\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMin, [Npp8u](#) \*pDeviceBuffer)  
*64-bit integer vector min method*
- [NppStatus nppsMinIndxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_16s.*
- [NppStatus nppsMinIndxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_32s.*
- [NppStatus nppsMinIndxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_32f.*
- [NppStatus nppsMinIndxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMinIndx\_64f.*
- [NppStatus nppsMinIndx\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit integer vector min index method*
- [NppStatus nppsMinIndx\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, int \*pIndx, [Npp8u](#) \*pDeviceBuffer)

*32-bit integer vector min index method*

- **NppStatus nppsMinIndx\_32f** (const **Npp32f** \*pSrc, int nLength, **Npp32f** \*pMin, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*32-bit float vector min index method*

- **NppStatus nppsMinIndx\_64f** (const **Npp64f** \*pSrc, int nLength, **Npp64f** \*pMin, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*64-bit float vector min index method*

- **NppStatus nppsMinAbsGetBufferSize\_16s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbs\_16s.*

- **NppStatus nppsMinAbsGetBufferSize\_32s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbs\_32s.*

- **NppStatus nppsMinAbs\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMinAbs, **Npp8u** \*pDeviceBuffer)

*16-bit integer vector min absolute method*

- **NppStatus nppsMinAbs\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMinAbs, **Npp8u** \*pDeviceBuffer)

*32-bit integer vector min absolute method*

- **NppStatus nppsMinAbsIndxGetBufferSize\_16s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbsIndx\_16s.*

- **NppStatus nppsMinAbsIndxGetBufferSize\_32s** (int nLength, int \*hpBufferSize)

*Device scratch buffer size (in bytes) for nppsMinAbsIndx\_32s.*

- **NppStatus nppsMinAbsIndx\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMinAbs, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*16-bit integer vector min absolute index method*

- **NppStatus nppsMinAbsIndx\_32s** (const **Npp32s** \*pSrc, int nLength, **Npp32s** \*pMinAbs, int \*pIndx, **Npp8u** \*pDeviceBuffer)

*32-bit integer vector min absolute index method*

## 7.186.1 Function Documentation

### 7.186.1.1 **NppStatus nppsMin\_16s** (const **Npp16s** \*pSrc, int nLength, **Npp16s** \*pMin, **Npp8u** \*pDeviceBuffer)

16-bit integer vector min method

#### Parameters:

**pSrc** Source Signal Pointer.

**nLength** Signal Length.

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.2 NppStatus nppsMin\_32f (const Npp32f \* pSrc, int nLength, Npp32f \* pMin, Npp8u \* pDeviceBuffer)**

32-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.3 NppStatus nppsMin\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMin, Npp8u \* pDeviceBuffer)**

32-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.4 NppStatus nppsMin\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMin, Npp8u \* pDeviceBuffer)**

64-bit integer vector min method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.5 NppStatus nppsMinAbs\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMinAbs, Npp8u \* pDeviceBuffer)**

16-bit integer vector min absolute method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMinAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.6 NppStatus nppsMinAbs\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMinAbs, Npp8u \* pDeviceBuffer)**

32-bit integer vector min absolute method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMinAbs* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.7 NppStatus nppsMinAbsGetBufferSize\_16s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMinAbs\_16s.

**Parameters:**

*nLength* [Signal Length](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.8 NppStatus nppsMinAbsGetBufferSize\_32s (int nLength, int \* hpBufferSize)**

Device scratch buffer size (in bytes) for nppsMinAbs\_32s.

**Parameters:**

**nLength** [Signal Length](#).

**hpBufferSize** Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.9 NppStatus nppsMinAbsIndx\_16s (const Npp16s \* pSrc, int nLength, Npp16s \* pMinAbs, int \* pIndx, Npp8u \* pDeviceBuffer)**

16-bit integer vector min absolute index method

**Parameters:**

**pSrc** [Source Signal Pointer](#).

**nLength** [Signal Length](#).

**pMinAbs** Pointer to the output result.

**pIndx** Pointer to the index value of the first minimum element.

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsIndxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.10 NppStatus nppsMinAbsIndx\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMinAbs, int \* pIndx, Npp8u \* pDeviceBuffer)**

32-bit integer vector min absolute index method

**Parameters:**

**pSrc** [Source Signal Pointer](#).

**nLength** [Signal Length](#).

**pMinAbs** Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinAbsIndxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.11 NppStatus nppsMinAbsIndxGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMinAbsIndx\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.12 NppStatus nppsMinAbsIndxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMinAbsIndx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.13 NppStatus nppsMinGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMin\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS



**7.186.1.14 NppStatus nppsMinGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMin\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.15 NppStatus nppsMinGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMin\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.16 NppStatus nppsMinGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMin\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.17 NppStatus nppsMinIndx\_16s (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMin*, int \* *pIndx*, Npp8u \* *pDeviceBuffer*)**

16-bit integer vector min index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIdxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.18** `NppStatus nppsMinIdx_32f (const Npp32f * pSrc, int nLength, Npp32f * pMin, int * pIndx, Npp8u * pDeviceBuffer)`

32-bit float vector min index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIdxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.186.1.19** `NppStatus nppsMinIdx_32s (const Npp32s * pSrc, int nLength, Npp32s * pMin, int * pIndx, Npp8u * pDeviceBuffer)`

32-bit integer vector min index method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIdxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.186.1.20 **NppStatus nppsMinIndx\_64f** (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pMin*, int \* *pIndx*, Npp8u \* *pDeviceBuffer*)

64-bit float vector min index method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the output result.

*pIndx* Pointer to the index value of the first minimum element.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinIndxGetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.186.1.21 **NppStatus nppsMinIndxGetBufferSize\_16s** (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMinIndx\_16s.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_SUCCESS

### 7.186.1.22 **NppStatus nppsMinIndxGetBufferSize\_32f** (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMinIndx\_32f.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_SUCCESS

**7.186.1.23 NppStatus nppsMinIndxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMinIndx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.186.1.24 NppStatus nppsMinIndxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMinIndx\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.187 Mean

### Functions

- [NppStatus nppsMeanGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_32f.*
- [NppStatus nppsMeanGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_32fc.*
- [NppStatus nppsMeanGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_64f.*
- [NppStatus nppsMeanGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_64fc.*
- [NppStatus nppsMeanGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_16s\_Sfs.*
- [NppStatus nppsMeanGetBufferSize\\_32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_32s\_Sfs.*
- [NppStatus nppsMeanGetBufferSize\\_16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMean\_16sc\_Sfs.*
- [NppStatus nppsMean\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector mean method*
- [NppStatus nppsMean\\_32fc](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp32fc](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector mean method*
- [NppStatus nppsMean\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double vector mean method*
- [NppStatus nppsMean\\_64fc](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64fc](#) \*pMean, [Npp8u](#) \*pDeviceBuffer)  
*64-bit double complex vector mean method*
- [NppStatus nppsMean\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMean, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit short vector mean with integer scaling method*
- [NppStatus nppsMean\\_32s\\_Sfs](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMean, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit integer vector mean with integer scaling method*
- [NppStatus nppsMean\\_16sc\\_Sfs](#) (const [Npp16sc](#) \*pSrc, int nLength, [Npp16sc](#) \*pMean, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

*16-bit short complex vector mean with integer scaling method*

### 7.187.1 Function Documentation

#### 7.187.1.1 **NppStatus nppsMean\_16s\_Sfs** (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMean*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit short vector mean with integer scaling method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.187.1.2 **NppStatus nppsMean\_16sc\_Sfs** (const Npp16sc \* *pSrc*, int *nLength*, Npp16sc \* *pMean*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit short complex vector mean with integer scaling method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_16sc\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.187.1.3 **NppStatus nppsMean\_32f** (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pMean*, Npp8u \* *pDeviceBuffer*)

32-bit float vector mean method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.187.1.4** `NppStatus nppsMean_32fc (const Npp32fc * pSrc, int nLength, Npp32fc * pMean, Npp8u * pDeviceBuffer)`

32-bit float complex vector mean method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.187.1.5** `NppStatus nppsMean_32s_Sfs (const Npp32s * pSrc, int nLength, Npp32s * pMean, int nScaleFactor, Npp8u * pDeviceBuffer)`

32-bit integer vector mean with integer scaling method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_32s\\_Sfs](#) to determine the minium number of bytes required.

*nScaleFactor* [Integer Result Scaling](#).

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.187.1.6 NppStatus nppsMean\_64f (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pMean*, Npp8u \* *pDeviceBuffer*)

64-bit double vector mean method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.187.1.7 NppStatus nppsMean\_64fc (const Npp64fc \* *pSrc*, int *nLength*, Npp64fc \* *pMean*, Npp8u \* *pDeviceBuffer*)

64-bit double complex vector mean method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.187.1.8 NppStatus nppsMeanGetBufferSize\_16s\_Sfs (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMean\_16s\_Sfs.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

#### Returns:

NPP\_SUCCESS



**7.187.1.9 NppStatus nppsMeanGetBufferSize\_16sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_16sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.187.1.10 NppStatus nppsMeanGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.187.1.11 NppStatus nppsMeanGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.187.1.12 NppStatus nppsMeanGetBufferSize\_32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.187.1.13 NppStatus nppsMeanGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.187.1.14 NppStatus nppsMeanGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMean\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.188 Standard Deviation

### Functions

- [NppStatus nppsStdDevGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_32f.*
- [NppStatus nppsStdDevGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_64f.*
- [NppStatus nppsStdDevGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_16s32s\_Sfs.*
- [NppStatus nppsStdDevGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsStdDev\_16s\_Sfs.*
- [NppStatus nppsStdDev\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector standard deviation method*
- [NppStatus nppsStdDev\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector standard deviation method*
- [NppStatus nppsStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector standard deviation method (return value is 32-bit)*
- [NppStatus nppsStdDev\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector standard deviation method (return value is also 16-bit)*

### 7.188.1 Function Documentation

#### 7.188.1.1 [NppStatus nppsStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit float vector standard deviation method (return value is 32-bit)

#### Parameters:

[pSrc](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pStdDev](#) Pointer to the output result.

[nScaleFactor](#) [Integer Result Scaling](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_16s32s\\_Sfs](#) to determine the minimum number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.188.1.2 NppStatus nppsStdDev\_16s\_Sfs (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pStdDev*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)**

16-bit float vector standard deviation method (return value is also 16-bit)

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pStdDev* Pointer to the output result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.188.1.3 NppStatus nppsStdDev\_32f (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pStdDev*, Npp8u \* *pDeviceBuffer*)**

32-bit float vector standard deviation method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pStdDev* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.188.1.4 NppStatus nppsStdDev\_64f (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pStdDev*, Npp8u \* *pDeviceBuffer*)**

64-bit float vector standard deviation method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pStdDev* Pointer to the output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsStdDevGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.188.1.5 NppStatus nppsStdDevGetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.188.1.6 NppStatus nppsStdDevGetBufferSize\_16s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.188.1.7 NppStatus nppsStdDevGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

**7.188.1.8 NppStatus nppsStdDevGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsStdDev\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.189 Mean And Standard Deviation

### Functions

- [NppStatus nppsMeanStdDevGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_32f.*
- [NppStatus nppsMeanStdDevGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_64f.*
- [NppStatus nppsMeanStdDevGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_16s32s\_Sfs.*
- [NppStatus nppsMeanStdDevGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device scratch buffer size (in bytes) for nppsMeanStdDev\_16s\_Sfs.*
- [NppStatus nppsMeanStdDev\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMean, [Npp32f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector mean and standard deviation method*
- [NppStatus nppsMeanStdDev\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMean, [Npp64f](#) \*pStdDev, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector mean and standard deviation method*
- [NppStatus nppsMeanStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pMean, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector mean and standard deviation method (return values are 32-bit)*
- [NppStatus nppsMeanStdDev\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMean, [Npp16s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit float vector mean and standard deviation method (return values are also 16-bit)*

### 7.189.1 Function Documentation

#### 7.189.1.1 [NppStatus nppsMeanStdDev\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pMean, [Npp32s](#) \*pStdDev, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

16-bit float vector mean and standard deviation method (return values are 32-bit)

#### Parameters:

**pSrc** [Source Signal Pointer](#).

**nLength** [Signal Length](#).

**pMean** Pointer to the output mean value.

**pStdDev** Pointer to the output standard deviation value.

**nScaleFactor** [Integer Result Scaling](#).

**pDeviceBuffer** Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_16s32s\\_Sfs](#) to determine the minimum number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.189.1.2 NppStatus nppsMeanStdDev\_16s\_Sfs (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMean*, Npp16s \* *pStdDev*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)**

16-bit float vector mean and standard deviation method (return values are also 16-bit)

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output mean value.

*pStdDev* Pointer to the output standard deviation value.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.189.1.3 NppStatus nppsMeanStdDev\_32f (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pMean*, Npp32f \* *pStdDev*, Npp8u \* *pDeviceBuffer*)**

32-bit float vector mean and standard deviation method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output mean value.

*pStdDev* Pointer to the output standard deviation value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.189.1.4 NppStatus nppsMeanStdDev\_64f (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pMean*, Npp64f \* *pStdDev*, Npp8u \* *pDeviceBuffer*)**

64-bit float vector mean and standard deviation method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMean* Pointer to the output mean value.

*pStdDev* Pointer to the output standard deviation value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMeanStdDevGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.189.1.5 NppStatus nppsMeanStdDevGetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMeanStdDev\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

#### 7.189.1.6 NppStatus nppsMeanStdDevGetBufferSize\_16s\_Sfs (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMeanStdDev\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

#### 7.189.1.7 NppStatus nppsMeanStdDevGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)

Device scratch buffer size (in bytes) for nppsMeanStdDev\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*. [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS



**7.189.1.8 NppStatus nppsMeanStdDevGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device scratch buffer size (in bytes) for nppsMeanStdDev\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*: [Scratch Buffer and Host Pointer](#).

**Returns:**

NPP\_SUCCESS

## 7.190 Minimum\_Maximum

### Functions

- [NppStatus nppsMinMaxGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_8u.*
- [NppStatus nppsMinMaxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_16s.*
- [NppStatus nppsMinMaxGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_16u.*
- [NppStatus nppsMinMaxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32s.*
- [NppStatus nppsMinMaxGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32u.*
- [NppStatus nppsMinMaxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_32f.*
- [NppStatus nppsMinMaxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMax\_64f.*
- [NppStatus nppsMinMax\\_8u](#) (const [Npp8u](#) \*pSrc, int nLength, [Npp8u](#) \*pMin, [Npp8u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*8-bit char vector min and max method*
- [NppStatus nppsMinMax\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, [Npp16s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short vector min and max method*
- [NppStatus nppsMinMax\\_16u](#) (const [Npp16u](#) \*pSrc, int nLength, [Npp16u](#) \*pMin, [Npp16u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*16-bit unsigned short vector min and max method*
- [NppStatus nppsMinMax\\_32u](#) (const [Npp32u](#) \*pSrc, int nLength, [Npp32u](#) \*pMin, [Npp32u](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit unsigned int vector min and max method*
- [NppStatus nppsMinMax\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, [Npp32s](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed int vector min and max method*
- [NppStatus nppsMinMax\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMin, [Npp32f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector min and max method*
- [NppStatus nppsMinMax\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMin, [Npp64f](#) \*pMax, [Npp8u](#) \*pDeviceBuffer)

*64-bit double vector min and max method*

- [NppStatus nppsMinMaxIdxGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_8u.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_16s.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_16u.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_32s.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_32u.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_32f.*
- [NppStatus nppsMinMaxIdxGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMinMaxIdx\_64f.*
- [NppStatus nppsMinMaxIdx\\_8u](#) (const [Npp8u](#) \*pSrc, int nLength, [Npp8u](#) \*pMin, int \*pMinIdx, [Npp8u](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*8-bit char vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_16s](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp16s](#) \*pMin, int \*pMinIdx, [Npp16s](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_16u](#) (const [Npp16u](#) \*pSrc, int nLength, [Npp16u](#) \*pMin, int \*pMinIdx, [Npp16u](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*16-bit unsigned short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, [Npp32s](#) \*pMin, int \*pMinIdx, [Npp32s](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_32u](#) (const [Npp32u](#) \*pSrc, int nLength, [Npp32u](#) \*pMin, int \*pMinIdx, [Npp32u](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*32-bit unsigned short vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pMin, int \*pMinIdx, [Npp32f](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector min and max with indices method*
- [NppStatus nppsMinMaxIdx\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pMin, int \*pMinIdx, [Npp64f](#) \*pMax, int \*pMaxIdx, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector min and max with indices method*

## 7.190.1 Function Documentation

### 7.190.1.1 `NppStatus nppsMinMax_16s (const Npp16s * pSrc, int nLength, Npp16s * pMin, Npp16s * pMax, Npp8u * pDeviceBuffer)`

16-bit signed short vector min and max method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.190.1.2 `NppStatus nppsMinMax_16u (const Npp16u * pSrc, int nLength, Npp16u * pMin, Npp16u * pMax, Npp8u * pDeviceBuffer)`

16-bit unsigned short vector min and max method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_16u](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.190.1.3 `NppStatus nppsMinMax_32f (const Npp32f * pSrc, int nLength, Npp32f * pMin, Npp32f * pMax, Npp8u * pDeviceBuffer)`

32-bit float vector min and max method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.190.1.4 NppStatus nppsMinMax\_32s (const Npp32s \* pSrc, int nLength, Npp32s \* pMin, Npp32s \* pMax, Npp8u \* pDeviceBuffer)

32-bit signed int vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.190.1.5 NppStatus nppsMinMax\_32u (const Npp32u \* pSrc, int nLength, Npp32u \* pMin, Npp32u \* pMax, Npp8u \* pDeviceBuffer)

32-bit unsigned int vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_32u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.190.1.6 NppStatus nppsMinMax\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pMin, Npp64f \* pMax, Npp8u \* pDeviceBuffer)

64-bit double vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.190.1.7** `NppStatus nppsMinMax_8u (const Npp8u * pSrc, int nLength, Npp8u * pMin, Npp8u * pMax, Npp8u * pDeviceBuffer)`

8-bit char vector min and max method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMax* Pointer to the max output result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxGetBufferSize\\_8u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.190.1.8** `NppStatus nppsMinMaxGetBufferSize_16s (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsMinMax\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.9** `NppStatus nppsMinMaxGetBufferSize_16u (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsMinMax\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.10 NppStatus nppsMinMaxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsMinMax\_32f*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.11 NppStatus nppsMinMaxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsMinMax\_32s*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.12 NppStatus nppsMinMaxGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsMinMax\_32u*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.13 NppStatus nppsMinMaxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.14 NppStatus nppsMinMaxGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMax\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.15 NppStatus nppsMinMaxIndx\_16s (const Npp16s \* *pSrc*, int *nLength*, Npp16s \* *pMin*, int \* *pMinIndx*, Npp16s \* *pMax*, int \* *pMaxIndx*, Npp8u \* *pDeviceBuffer*)**

16-bit signed short vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_16s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



### 7.190.1.16 **NppStatus nppsMinMaxIndx\_16u** (const Npp16u \* *pSrc*, int *nLength*, Npp16u \* *pMin*, int \* *pMinIndx*, Npp16u \* *pMax*, int \* *pMaxIndx*, Npp8u \* *pDeviceBuffer*)

16-bit unsigned short vector min and max with indices method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_16u](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.190.1.17 **NppStatus nppsMinMaxIndx\_32f** (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pMin*, int \* *pMinIndx*, Npp32f \* *pMax*, int \* *pMaxIndx*, Npp8u \* *pDeviceBuffer*)

32-bit float vector min and max with indices method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.190.1.18 **NppStatus nppsMinMaxIndx\_32s** (const Npp32s \* *pSrc*, int *nLength*, Npp32s \* *pMin*, int \* *pMinIndx*, Npp32s \* *pMax*, int \* *pMaxIndx*, Npp8u \* *pDeviceBuffer*)

32-bit signed short vector min and max with indices method

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.190.1.19** `NppStatus nppsMinMaxIndx_32u (const Npp32u * pSrc, int nLength, Npp32u * pMin, int * pMinIndx, Npp32u * pMax, int * pMaxIndx, Npp8u * pDeviceBuffer)`

32-bit unsigned short vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_32u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.190.1.20** `NppStatus nppsMinMaxIndx_64f (const Npp64f * pSrc, int nLength, Npp64f * pMin, int * pMinIndx, Npp64f * pMax, int * pMaxIndx, Npp8u * pDeviceBuffer)`

64-bit float vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.190.1.21** `NppStatus nppsMinMaxIndx_8u (const Npp8u * pSrc, int nLength, Npp8u * pMin, int * pMinIndx, Npp8u * pMax, int * pMaxIndx, Npp8u * pDeviceBuffer)`

8-bit char vector min and max with indices method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pMin* Pointer to the min output result.

*pMinIndx* Pointer to the index of the first min value.

*pMax* Pointer to the max output result.

*pMaxIndx* Pointer to the index of the first max value.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMinMaxIndxGetBufferSize\\_8u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.190.1.22** `NppStatus nppsMinMaxIndxGetBufferSize_16s (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsMinMaxIndx\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.23** `NppStatus nppsMinMaxIndxGetBufferSize_16u (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsMinMaxIndx\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.24 NppStatus nppsMinMaxIdxGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.25 NppStatus nppsMinMaxIdxGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.26 NppStatus nppsMinMaxIdxGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_32u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.27 NppStatus nppsMinMaxIdxGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIdx\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.190.1.28 NppStatus nppsMinMaxIndxGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMinMaxIndx\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.191 Infinity Norm

### Functions

- [NppStatus nppsNormInfGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_32f.*
- [NppStatus nppsNorm\\_Inf\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector C norm method*
- [NppStatus nppsNormInfGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_64f.*
- [NppStatus nppsNorm\\_Inf\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector C norm method*
- [NppStatus nppsNormInfGetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32f.*
- [NppStatus nppsNorm\\_Inf\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector C norm method, return value is 32-bit float.*
- [NppStatus nppsNormInfGetBufferSize\\_32fc32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_32fc32f.*
- [NppStatus nppsNorm\\_Inf\\_32fc32f](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector C norm method, return value is 32-bit float.*
- [NppStatus nppsNormInfGetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_64fc64f.*
- [NppStatus nppsNorm\\_Inf\\_64fc64f](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex vector C norm method, return value is 64-bit float.*
- [NppStatus nppsNormInfGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32s\_Sfs.*
- [NppStatus nppsNorm\\_Inf\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector C norm method, return value is 32-bit signed integer.*

### 7.191.1 Function Documentation

#### 7.191.1.1 `NppStatus nppsNorm_Inf_16s32f (const Npp16s * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector C norm method, return value is 32-bit float.

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.191.1.2 `NppStatus nppsNorm_Inf_16s32s_Sfs (const Npp16s * pSrc, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer vector C norm method, return value is 32-bit signed integer.

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.191.1.3 `NppStatus nppsNorm_Inf_32f (const Npp32f * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float vector C norm method

##### Parameters:

*pSrc* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.191.1.4 **NppStatus nppsNorm\_Inf\_32fc32f** (const Npp32fc \* *pSrc*, int *nLength*, Npp32f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

32-bit float complex vector C norm method, return value is 32-bit float.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_32fc32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.191.1.5 **NppStatus nppsNorm\_Inf\_64f** (const Npp64f \* *pSrc*, int *nLength*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

64-bit float vector C norm method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_64f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.191.1.6 **NppStatus nppsNorm\_Inf\_64fc64f** (const Npp64fc \* *pSrc*, int *nLength*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

64-bit float complex vector C norm method, return value is 64-bit float.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormInfGetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.191.1.7 NppStatus nppsNormInfGetBufferSize\_16s32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.191.1.8 NppStatus nppsNormInfGetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.191.1.9 NppStatus nppsNormInfGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.191.1.10 NppStatus nppsNormInfGetBufferSize\_32fc32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_32fc32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.191.1.11 NppStatus nppsNormInfGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.191.1.12 NppStatus nppsNormInfGetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_Inf\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.192 L1 Norm

### Functions

- [NppStatus nppsNormL1GetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_32f.*
- [NppStatus nppsNorm\\_L1\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector L1 norm method*
- [NppStatus nppsNormL1GetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_64f.*
- [NppStatus nppsNorm\\_L1\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector L1 norm method*
- [NppStatus nppsNormL1GetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_16s32f.*
- [NppStatus nppsNorm\\_L1\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L1 norm method, return value is 32-bit float.*
- [NppStatus nppsNormL1GetBufferSize\\_32fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_32fc64f.*
- [NppStatus nppsNorm\\_L1\\_32fc64f](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector L1 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL1GetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_64fc64f.*
- [NppStatus nppsNorm\\_L1\\_64fc64f](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex vector L1 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL1GetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_16s32s\_Sfs.*
- [NppStatus nppsNorm\\_L1\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L1 norm method, return value is 32-bit signed integer.*
- [NppStatus nppsNormL1GetBufferSize\\_16s64s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L1\_16s64s\_Sfs.*
- [NppStatus nppsNorm\\_L1\\_16s64s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp64s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

*16-bit signed short integer vector L1 norm method, return value is 64-bit signed integer.*

## 7.192.1 Function Documentation

### 7.192.1.1 NppStatus nppsNorm\_L1\_16s32f (const Npp16s \* pSrc, int nLength, Npp32f \* pNorm, Npp8u \* pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 32-bit float.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the L1 norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.192.1.2 NppStatus nppsNorm\_L1\_16s32s\_Sfs (const Npp16s \* pSrc, int nLength, Npp32s \* pNorm, int nScaleFactor, Npp8u \* pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 32-bit signed integer.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.192.1.3 NppStatus nppsNorm\_L1\_16s64s\_Sfs (const Npp16s \* pSrc, int nLength, Npp64s \* pNorm, int nScaleFactor, Npp8u \* pDeviceBuffer)

16-bit signed short integer vector L1 norm method, return value is 64-bit signed integer.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_16s64s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.192.1.4** `NppStatus nppsNorm_L1_32f (const Npp32f * pSrc, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float vector L1 norm method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.192.1.5** `NppStatus nppsNorm_L1_32fc64f (const Npp32fc * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float complex vector L1 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_32fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.192.1.6** `NppStatus nppsNorm_L1_64f (const Npp64f * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float vector L1 norm method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.192.1.7** `NppStatus nppsNorm_L1_64fc64f (const Npp64fc * pSrc, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex vector L1 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL1GetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.192.1.8** `NppStatus nppsNormL1GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm\_L1\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.192.1.9** `NppStatus nppsNormL1GetBufferSize_16s32s_Sfs (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for nppsNorm\_L1\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.192.1.10 NppStatus nppsNormL1GetBufferSize\_16s64s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L1\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.192.1.11 NppStatus nppsNormL1GetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L1\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.192.1.12 NppStatus nppsNormL1GetBufferSize\_32fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L1\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.192.1.13 NppStatus nppsNormL1GetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L1\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.192.1.14 NppStatus nppsNormL1GetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L1\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS



## 7.193 L2 Norm

### Functions

- [NppStatus nppsNormL2GetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_32f.*
- [NppStatus nppsNorm\\_L2\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float vector L2 norm method*
- [NppStatus nppsNormL2GetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_64f.*
- [NppStatus nppsNorm\\_L2\\_64f](#) (const [Npp64f](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float vector L2 norm method*
- [NppStatus nppsNormL2GetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_16s32f.*
- [NppStatus nppsNorm\\_L2\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L2 norm method, return value is 32-bit float.*
- [NppStatus nppsNormL2GetBufferSize\\_32fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_32fc64f.*
- [NppStatus nppsNorm\\_L2\\_32fc64f](#) (const [Npp32fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex vector L2 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL2GetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_64fc64f.*
- [NppStatus nppsNorm\\_L2\\_64fc64f](#) (const [Npp64fc](#) \*pSrc, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex vector L2 norm method, return value is 64-bit float.*
- [NppStatus nppsNormL2GetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2\_16s32s\_Sfs.*
- [NppStatus nppsNorm\\_L2\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer vector L2 norm method, return value is 32-bit signed integer.*
- [NppStatus nppsNormL2SqrGetBufferSize\\_16s64s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNorm\_L2Sqr\_16s64s\_Sfs.*
- [NppStatus nppsNorm\\_L2Sqr\\_16s64s\\_Sfs](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp64s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

*16-bit signed short integer vector L2 Square norm method, return value is 64-bit signed integer.*

### 7.193.1 Function Documentation

#### 7.193.1.1 **NppStatus nppsNorm\_L2\_16s32f** (const Npp16s \* *pSrc*, int *nLength*, Npp32f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer vector L2 norm method, return value is 32-bit float.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL2GetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.193.1.2 **NppStatus nppsNorm\_L2\_16s32s\_Sfs** (const Npp16s \* *pSrc*, int *nLength*, Npp32s \* *pNorm*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer vector L2 norm method, return value is 32-bit signed integer.

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL2GetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.193.1.3 **NppStatus nppsNorm\_L2\_32f** (const Npp32f \* *pSrc*, int *nLength*, Npp32f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

32-bit float vector L2 norm method

##### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.193.1.4 NppStatus nppsNorm\_L2\_32fc64f (const Npp32fc \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

32-bit float complex vector L2 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize\\_32fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.193.1.5 NppStatus nppsNorm\_L2\_64f (const Npp64f \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float vector L2 norm method

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).

Use [nppsNormL2GetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.193.1.6 NppStatus nppsNorm\_L2\_64fc64f (const Npp64fc \* pSrc, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float complex vector L2 norm method, return value is 64-bit float.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL2GetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.193.1.7 NppStatus nppsNorm\_L2Sqr\_16s64s\_Sfs (const Npp16s \* pSrc, int nLength, Npp64s \* pNorm, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer vector L2 Square norm method, return value is 64-bit signed integer.

**Parameters:**

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormL2SqrGetBufferSize\\_16s64s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.193.1.8 NppStatus nppsNormL2GetBufferSize\_16s32f (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L2\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.193.1.9 NppStatus nppsNormL2GetBufferSize\_16s32s\_Sfs (int nLength, int \* hpBufferSize)**

Device-buffer size (in bytes) for nppsNorm\_L2\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.193.1.10 NppStatus nppsNormL2GetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L2\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.193.1.11 NppStatus nppsNormL2GetBufferSize\_32fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L2\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.193.1.12 NppStatus nppsNormL2GetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L2\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.193.1.13 NppStatus nppsNormL2GetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNorm\_L2\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.193.1.14** **NppStatus nppsNormL2SqrGetBufferSize\_16s64s\_Sfs** (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsNorm\_L2Sqr\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.194 Infinity Norm Diff

### Functions

- [NppStatus nppsNormDiffInfGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_32f.*
- [NppStatus nppsNormDiff\\_Inf\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float C norm method on two vectors' difference*
- [NppStatus nppsNormDiffInfGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_64f.*
- [NppStatus nppsNormDiff\\_Inf\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float C norm method on two vectors' difference*
- [NppStatus nppsNormDiffInfGetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_16s32f.*
- [NppStatus nppsNormDiff\\_Inf\\_16s32f](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit float.*
- [NppStatus nppsNormDiffInfGetBufferSize\\_32fc32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_32fc32f.*
- [NppStatus nppsNormDiff\\_Inf\\_32fc32f](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex C norm method on two vectors' difference, return value is 32-bit float.*
- [NppStatus nppsNormDiffInfGetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_64fc64f.*
- [NppStatus nppsNormDiff\\_Inf\\_64fc64f](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex C norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffInfGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_Inf\_16s32s\_Sfs.*
- [NppStatus nppsNormDiff\\_Inf\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit signed integer.*

## 7.194.1 Function Documentation

### 7.194.1.1 `NppStatus nppsNormDiff_Inf_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit float.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.194.1.2 `NppStatus nppsNormDiff_Inf_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer C norm method on two vectors' difference, return value is 32-bit signed integer.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.194.1.3 `NppStatus nppsNormDiff_Inf_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float C norm method on two vectors' difference

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).



*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.194.1.4 NppStatus nppsNormDiff\_Inf\_32fc32f (const Npp32fc \* pSrc1, const Npp32fc \* pSrc2, int nLength, Npp32f \* pNorm, Npp8u \* pDeviceBuffer)**

32-bit float complex C norm method on two vectors' difference, return value is 32-bit float.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_32fc32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.194.1.5 NppStatus nppsNormDiff\_Inf\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, int nLength, Npp64f \* pNorm, Npp8u \* pDeviceBuffer)**

64-bit float C norm method on two vectors' difference

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.194.1.6 **NppStatus nppsNormDiff\_Inf\_64fc64f** (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, int *nLength*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

64-bit float complex C norm method on two vectors' difference, return value is 64-bit float.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffInfGetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.194.1.7 **NppStatus nppsNormDiffInfGetBufferSize\_16s32f** (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff\_Inf\_16s32f.

##### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

##### Returns:

NPP\_SUCCESS

#### 7.194.1.8 **NppStatus nppsNormDiffInfGetBufferSize\_16s32s\_Sfs** (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff\_Inf\_16s32s\_Sfs.

##### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

##### Returns:

NPP\_SUCCESS

#### 7.194.1.9 **NppStatus nppsNormDiffInfGetBufferSize\_32f** (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff\_Inf\_32f.

##### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.194.1.10 NppStatus nppsNormDiffInfGetBufferSize\_32fc32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for `nppsNormDiff_Inf_32fc32f`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.194.1.11 NppStatus nppsNormDiffInfGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for `nppsNormDiff_Inf_64f`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.194.1.12 NppStatus nppsNormDiffInfGetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for `nppsNormDiff_Inf_64fc64f`.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.195 L1 Norm Diff

### Functions

- **NppStatus nppsNormDiffL1GetBufferSize\_32f** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_32f.*
- **NppStatus nppsNormDiff\_L1\_32f** (const **Npp32f** \*pSrc1, const **Npp32f** \*pSrc2, int nLength, **Npp32f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*32-bit float L1 norm method on two vectors' difference*
- **NppStatus nppsNormDiffL1GetBufferSize\_64f** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_64f.*
- **NppStatus nppsNormDiff\_L1\_64f** (const **Npp64f** \*pSrc1, const **Npp64f** \*pSrc2, int nLength, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*64-bit float L1 norm method on two vectors' difference*
- **NppStatus nppsNormDiffL1GetBufferSize\_16s32f** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s32f.*
- **NppStatus nppsNormDiff\_L1\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, int nLength, **Npp32f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit float.*
- **NppStatus nppsNormDiffL1GetBufferSize\_32fc64f** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_32fc64f.*
- **NppStatus nppsNormDiff\_L1\_32fc64f** (const **Npp32fc** \*pSrc1, const **Npp32fc** \*pSrc2, int nLength, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*32-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.*
- **NppStatus nppsNormDiffL1GetBufferSize\_64fc64f** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_64fc64f.*
- **NppStatus nppsNormDiff\_L1\_64fc64f** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, int nLength, **Npp64f** \*pNorm, **Npp8u** \*pDeviceBuffer)  
*64-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.*
- **NppStatus nppsNormDiffL1GetBufferSize\_16s32s\_Sfs** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s32s\_Sfs.*
- **NppStatus nppsNormDiff\_L1\_16s32s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, int nLength, **Npp32s** \*pNorm, int nScaleFactor, **Npp8u** \*pDeviceBuffer)  
*16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit signed integer.*
- **NppStatus nppsNormDiffL1GetBufferSize\_16s64s\_Sfs** (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s64s\_Sfs.*
- **NppStatus nppsNormDiff\_L1\_16s64s\_Sfs** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, int nLength, **Npp64s** \*pNorm, int nScaleFactor, **Npp8u** \*pDeviceBuffer)

*16-bit signed short integer L1 norm method on two vectors' difference, return value is 64-bit signed integer.*

### 7.195.1 Function Documentation

**7.195.1.1** `NppStatus nppsNormDiff_L1_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit float.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the L1 norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.195.1.2** `NppStatus nppsNormDiff_L1_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 32-bit signed integer.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#)..

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.195.1.3** `NppStatus nppsNormDiff_L1_16s64s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L1 norm method on two vectors' difference, return value is 64-bit signed integer.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_16s64s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.195.1.4 **NppStatus nppsNormDiff\_L1\_32f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, int *nLength*, Npp32f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

32-bit float L1 norm method on two vectors' difference

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.195.1.5 **NppStatus nppsNormDiff\_L1\_32fc64f** (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

32-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_32fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.195.1.6 NppStatus nppsNormDiff\_L1\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, int *nLength*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

64-bit float L1 norm method on two vectors' difference

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.195.1.7 NppStatus nppsNormDiff\_L1\_64fc64f (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, int *nLength*, Npp64f \* *pNorm*, Npp8u \* *pDeviceBuffer*)

64-bit float complex L1 norm method on two vectors' difference, return value is 64-bit float.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL1GetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.195.1.8 NppStatus nppsNormDiffL1GetBufferSize\_16s32f (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s32f.

#### Parameters:

*nLength* Signal Length.

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

#### Returns:

NPP\_SUCCESS

**7.195.1.9 NppStatus nppsNormDiffL1GetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.195.1.10 NppStatus nppsNormDiffL1GetBufferSize\_16s64s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.195.1.11 NppStatus nppsNormDiffL1GetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.195.1.12 NppStatus nppsNormDiffL1GetBufferSize\_32fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS



**7.195.1.13 NppStatus nppsNormDiffL1GetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.195.1.14 NppStatus nppsNormDiffL1GetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L1\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.196 L2 Norm Diff

### Functions

- [NppStatus nppsNormDiffL2GetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_32f.*
- [NppStatus nppsNormDiff\\_L2\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float L2 norm method on two vectors' difference*
- [NppStatus nppsNormDiffL2GetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_64f.*
- [NppStatus nppsNormDiff\\_L2\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float L2 norm method on two vectors' difference*
- [NppStatus nppsNormDiffL2GetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_16s32f.*
- [NppStatus nppsNormDiff\\_L2\\_16s32f](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit float.*
- [NppStatus nppsNormDiffL2GetBufferSize\\_32fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_32fc64f.*
- [NppStatus nppsNormDiff\\_L2\\_32fc64f](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffL2GetBufferSize\\_64fc64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_64fc64f.*
- [NppStatus nppsNormDiff\\_L2\\_64fc64f](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pNorm, [Npp8u](#) \*pDeviceBuffer)  
*64-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.*
- [NppStatus nppsNormDiffL2GetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2\_16s32s\_Sfs.*
- [NppStatus nppsNormDiff\\_L2\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit signed integer.*
- [NppStatus nppsNormDiffL2SqrGetBufferSize\\_16s64s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsNormDiff\_L2Sqr\_16s64s\_Sfs.*
- [NppStatus nppsNormDiff\\_L2Sqr\\_16s64s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp64s](#) \*pNorm, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)

*16-bit signed short integer L2 Square norm method on two vectors' difference, return value is 64-bit signed integer.*

## 7.196.1 Function Documentation

### 7.196.1.1 `NppStatus nppsNormDiff_L2_16s32f (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit float.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.196.1.2 `NppStatus nppsNormDiff_L2_16s32s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp32s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L2 norm method on two vectors' difference, return value is 32-bit signed integer.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.196.1.3 `NppStatus nppsNormDiff_L2_32f (const Npp32f * pSrc1, const Npp32f * pSrc2, int nLength, Npp32f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float L2 norm method on two vectors' difference

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.196.1.4** `NppStatus nppsNormDiff_L2_32fc64f (const Npp32fc * pSrc1, const Npp32fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

32-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_32fc64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.196.1.5** `NppStatus nppsNormDiff_L2_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float L2 norm method on two vectors' difference

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.196.1.6 `NppStatus nppsNormDiff_L2_64fc64f (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pNorm, Npp8u * pDeviceBuffer)`

64-bit float complex L2 norm method on two vectors' difference, return value is 64-bit float.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2GetBufferSize\\_64fc64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.196.1.7 `NppStatus nppsNormDiff_L2Sqr_16s64s_Sfs (const Npp16s * pSrc1, const Npp16s * pSrc2, int nLength, Npp64s * pNorm, int nScaleFactor, Npp8u * pDeviceBuffer)`

16-bit signed short integer L2 Square norm method on two vectors' difference, return value is 64-bit signed integer.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pNorm* Pointer to the norm result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsNormDiffL2SqrGetBufferSize\\_16s64s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.196.1.8 `NppStatus nppsNormDiffL2GetBufferSize_16s32f (int nLength, int * hpBufferSize)`

Device-buffer size (in bytes) for `nppsNormDiff_L2_16s32f`.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

#### Returns:

NPP\_SUCCESS

**7.196.1.9 NppStatus nppsNormDiffL2GetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.196.1.10 NppStatus nppsNormDiffL2GetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.196.1.11 NppStatus nppsNormDiffL2GetBufferSize\_32fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_32fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.196.1.12 NppStatus nppsNormDiffL2GetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.196.1.13 NppStatus nppsNormDiffL2GetBufferSize\_64fc64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2\_64fc64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.196.1.14 NppStatus nppsNormDiffL2SqrGetBufferSize\_16s64s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsNormDiff\_L2Sqr\_16s64s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.197 Dot Product

### Functions

- [NppStatus nppsDotProdGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f.*
- [NppStatus nppsDotProd\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp32f](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float dot product method, return value is 32-bit float.*
- [NppStatus nppsDotProdGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32fc.*
- [NppStatus nppsDotProd\\_32fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp32fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex dot product method, return value is 32-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32f32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f32fc.*
- [NppStatus nppsDotProd\\_32f32fc](#) (const [Npp32f](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp32fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float and 32-bit float complex dot product method, return value is 32-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32f64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f64f.*
- [NppStatus nppsDotProd\\_32f64f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float dot product method, return value is 64-bit float.*
- [NppStatus nppsDotProdGetBufferSize\\_32fc64fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32fc64fc.*
- [NppStatus nppsDotProd\\_32fc64fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float complex dot product method, return value is 64-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32f32fc64fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32f32fc64fc.*
- [NppStatus nppsDotProd\\_32f32fc64fc](#) (const [Npp32f](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*32-bit float and 32-bit float complex dot product method, return value is 64-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_64f.*
- [NppStatus nppsDotProd\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)



64-bit float dot product method, return value is 64-bit float.

- **NppStatus nppsDotProdGetBufferSize\_64fc** (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for nppsDotProd\_64fc.
- **NppStatus nppsDotProd\_64fc** (const **Npp64fc** \*pSrc1, const **Npp64fc** \*pSrc2, int nLength, **Npp64fc** \*pDp, **Npp8u** \*pDeviceBuffer)  
64-bit float complex dot product method, return value is 64-bit float complex.
- **NppStatus nppsDotProdGetBufferSize\_64f64fc** (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for nppsDotProd\_64f64fc.
- **NppStatus nppsDotProd\_64f64fc** (const **Npp64f** \*pSrc1, const **Npp64fc** \*pSrc2, int nLength, **Npp64fc** \*pDp, **Npp8u** \*pDeviceBuffer)  
64-bit float and 64-bit float complex dot product method, return value is 64-bit float complex.
- **NppStatus nppsDotProdGetBufferSize\_16s64s** (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for nppsDotProd\_16s64s.
- **NppStatus nppsDotProd\_16s64s** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, int nLength, **Npp64s** \*pDp, **Npp8u** \*pDeviceBuffer)  
16-bit signed short integer dot product method, return value is 64-bit signed integer.
- **NppStatus nppsDotProdGetBufferSize\_16sc64sc** (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for nppsDotProd\_16sc64sc.
- **NppStatus nppsDotProd\_16sc64sc** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, int nLength, **Npp64sc** \*pDp, **Npp8u** \*pDeviceBuffer)  
16-bit signed short integer complex dot product method, return value is 64-bit signed integer complex.
- **NppStatus nppsDotProdGetBufferSize\_16s16sc64sc** (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for nppsDotProd\_16s16sc64sc.
- **NppStatus nppsDotProd\_16s16sc64sc** (const **Npp16s** \*pSrc1, const **Npp16sc** \*pSrc2, int nLength, **Npp64sc** \*pDp, **Npp8u** \*pDeviceBuffer)  
16-bit signed short integer and 16-bit signed short integer short dot product method, return value is 64-bit signed integer complex.
- **NppStatus nppsDotProdGetBufferSize\_16s32f** (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for nppsDotProd\_16s32f.
- **NppStatus nppsDotProd\_16s32f** (const **Npp16s** \*pSrc1, const **Npp16s** \*pSrc2, int nLength, **Npp32f** \*pDp, **Npp8u** \*pDeviceBuffer)  
16-bit signed short integer dot product method, return value is 32-bit float.
- **NppStatus nppsDotProdGetBufferSize\_16sc32fc** (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for nppsDotProd\_16sc32fc.
- **NppStatus nppsDotProd\_16sc32fc** (const **Npp16sc** \*pSrc1, const **Npp16sc** \*pSrc2, int nLength, **Npp32fc** \*pDp, **Npp8u** \*pDeviceBuffer)  
16-bit signed short integer complex dot product method, return value is 32-bit float complex.

- [NppStatus nppsDotProdGetBufferSize\\_16s16sc32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s16sc32fc.*
- [NppStatus nppsDotProd\\_16s16sc32fc](#) (const [Npp16s](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp32fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit float complex.*
- [NppStatus nppsDotProdGetBufferSize\\_16s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s\_Sfs.*
- [NppStatus nppsDotProd\\_16s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp16s](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer dot product method, return value is 16-bit signed short integer.*
- [NppStatus nppsDotProdGetBufferSize\\_16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16sc\_Sfs.*
- [NppStatus nppsDotProd\\_16sc\\_Sfs](#) (const [Npp16sc](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp16sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32s\_Sfs.*
- [NppStatus nppsDotProd\\_32s\\_Sfs](#) (const [Npp32s](#) \*pSrc1, const [Npp32s](#) \*pSrc2, int nLength, [Npp32s](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed integer dot product method, return value is 32-bit signed integer.*
- [NppStatus nppsDotProdGetBufferSize\\_32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32sc\_Sfs.*
- [NppStatus nppsDotProd\\_32sc\\_Sfs](#) (const [Npp32sc](#) \*pSrc1, const [Npp32sc](#) \*pSrc2, int nLength, [Npp32sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed integer complex dot product method, return value is 32-bit signed integer complex.*
- [NppStatus nppsDotProdGetBufferSize\\_16s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s32s\_Sfs.*
- [NppStatus nppsDotProd\\_16s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp32s](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer dot product method, return value is 32-bit signed integer.*
- [NppStatus nppsDotProdGetBufferSize\\_16s16sc32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s16sc32sc\_Sfs.*
- [NppStatus nppsDotProd\\_16s16sc32sc\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp32sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.*

- [NppStatus nppsDotProdGetBufferSize\\_16s32s32s\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s32s32s\_Sfs.*
- [NppStatus nppsDotProd\\_16s32s32s\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp32s](#) \*pSrc2, int nLength, [Npp32s](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer and 32-bit signed integer dot product method, return value is 32-bit signed integer.*
- [NppStatus nppsDotProdGetBufferSize\\_16s16sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16s16sc\_Sfs.*
- [NppStatus nppsDotProd\\_16s16sc\\_Sfs](#) (const [Npp16s](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp16sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.*
- [NppStatus nppsDotProdGetBufferSize\\_16sc32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_16sc32sc\_Sfs.*
- [NppStatus nppsDotProd\\_16sc32sc\\_Sfs](#) (const [Npp16sc](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp32sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.*
- [NppStatus nppsDotProdGetBufferSize\\_32s32sc\\_Sfs](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsDotProd\_32s32sc\_Sfs.*
- [NppStatus nppsDotProd\\_32s32sc\\_Sfs](#) (const [Npp32s](#) \*pSrc1, const [Npp32sc](#) \*pSrc2, int nLength, [Npp32sc](#) \*pDp, int nScaleFactor, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed short integer and 32-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.*

## 7.197.1 Function Documentation

### 7.197.1.1 [NppStatus nppsDotProd\\_16s16sc32fc](#) (const [Npp16s](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp32fc](#) \*pDp, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit float complex.

#### Parameters:

[pSrc1](#) [Source Signal Pointer](#).

[pSrc2](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pDp](#) Pointer to the dot product result.

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s16sc32fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.2 NppStatus nppsDotProd\_16s16sc32sc\_Sfs (const Npp16s \* *pSrc1*, const Npp16sc \* *pSrc2*, int *nLength*, Npp32sc \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s16sc32sc\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.3 NppStatus nppsDotProd\_16s16sc64sc (const Npp16s \* *pSrc1*, const Npp16sc \* *pSrc2*, int *nLength*, Npp64sc \* *pDp*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer and 16-bit signed short integer short dot product method, return value is 64-bit signed integer complex.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s16sc64sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.4 NppStatus nppsDotProd\_16s16sc\_Sfs (const Npp16s \* *pSrc1*, const Npp16sc \* *pSrc2*, int *nLength*, Npp16sc \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer and 16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*nScaleFactor* Integer Result Scaling.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s16sc\\_Sfs](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.5 NppStatus nppsDotProd\_16s32f (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, int *nLength*, Npp32f \* *pDp*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer dot product method, return value is 32-bit float.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.6 NppStatus nppsDotProd\_16s32s32s\_Sfs (const Npp16s \* *pSrc1*, const Npp32s \* *pSrc2*, int *nLength*, Npp32s \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer and 32-bit signed integer dot product method, return value is 32-bit signed integer.

##### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s32s32s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.197.1.7** `NppStatus nppsDotProd_16s32s_Sfs (const Npp16s *pSrc1, const Npp16s *pSrc2, int nLength, Npp32s *pDp, int nScaleFactor, Npp8u *pDeviceBuffer)`

16-bit signed short integer dot product method, return value is 32-bit signed integer.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s32s\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.197.1.8** `NppStatus nppsDotProd_16s64s (const Npp16s *pSrc1, const Npp16s *pSrc2, int nLength, Npp64s *pDp, Npp8u *pDeviceBuffer)`

16-bit signed short integer dot product method, return value is 64-bit signed integer.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s64s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.9 **NppStatus nppsDotProd\_16s\_Sfs** (const Npp16s \* *pSrc1*, const Npp16s \* *pSrc2*, int *nLength*, Npp16s \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer dot product method, return value is 16-bit signed short integer.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.10 **NppStatus nppsDotProd\_16sc32fc** (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, int *nLength*, Npp32fc \* *pDp*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer complex dot product method, return value is 32-bit float complex.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc32fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.11 **NppStatus nppsDotProd\_16sc32sc\_Sfs** (const Npp16sc \* *pSrc1*, const Npp16sc \* *pSrc2*, int *nLength*, Npp32sc \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

16-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc32sc\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.197.1.12 NppStatus nppsDotProd\_16sc64sc (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp64sc \* pDp, Npp8u \* pDeviceBuffer)**

16-bit signed short integer complex dot product method, return value is 64-bit signed integer complex.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc64sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.197.1.13 NppStatus nppsDotProd\_16sc\_Sfs (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp16sc \* pDp, int nScaleFactor, Npp8u \* pDeviceBuffer)**

16-bit signed short integer complex dot product method, return value is 16-bit signed short integer complex.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_16sc\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



#### 7.197.1.14 **NppStatus nppsDotProd\_32f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, int *nLength*, Npp32f \* *pDp*, Npp8u \* *pDeviceBuffer*)

32-bit float dot product method, return value is 32-bit float.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.15 **NppStatus nppsDotProd\_32f32fc** (const Npp32f \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp32fc \* *pDp*, Npp8u \* *pDeviceBuffer*)

32-bit float and 32-bit float complex dot product method, return value is 32-bit float complex.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f32fc](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.16 **NppStatus nppsDotProd\_32f32fc64fc** (const Npp32f \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp64fc \* *pDp*, Npp8u \* *pDeviceBuffer*)

32-bit float and 32-bit float complex dot product method, return value is 64-bit float complex.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f32fc64fc](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.17 **NppStatus nppsDotProd\_32f64f** (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, int *nLength*, Npp64f \* *pDp*, Npp8u \* *pDeviceBuffer*)

32-bit float dot product method, return value is 64-bit float.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32f64f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.18 **NppStatus nppsDotProd\_32fc** (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp32fc \* *pDp*, Npp8u \* *pDeviceBuffer*)

32-bit float complex dot product method, return value is 32-bit float complex.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.197.1.19 **NppStatus nppsDotProd\_32fc64fc** (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp64fc \* *pDp*, Npp8u \* *pDeviceBuffer*)

32-bit float complex dot product method, return value is 64-bit float complex.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*pDeviceBuffer* [Pointer to the required device memory allocation](#), [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32fc64fc](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.20 NppStatus nppsDotProd\_32s32sc\_Sfs (const Npp32s \* *pSrc1*, const Npp32sc \* *pSrc2*, int *nLength*, Npp32sc \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

32-bit signed short integer and 32-bit signed short integer complex dot product method, return value is 32-bit signed integer complex.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32s32sc\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.21 NppStatus nppsDotProd\_32s\_Sfs (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, int *nLength*, Npp32s \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

32-bit signed integer dot product method, return value is 32-bit signed integer.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* [Pointer to the dot product result](#).

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* [Pointer to the required device memory allocation, Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32s\\_Sfs](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.22 NppStatus nppsDotProd\_32sc\_Sfs (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, int *nLength*, Npp32sc \* *pDp*, int *nScaleFactor*, Npp8u \* *pDeviceBuffer*)

32-bit signed integer complex dot product method, return value is 32-bit signed integer complex.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*nScaleFactor* [Integer Result Scaling](#).

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_32sc\\_Sfs](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.197.1.23 NppStatus nppsDotProd\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, int nLength, Npp64f \* pDp, Npp8u \* pDeviceBuffer)**

64-bit float dot product method, return value is 64-bit float.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.197.1.24 NppStatus nppsDotProd\_64f64fc (const Npp64f \* pSrc1, const Npp64fc \* pSrc2, int nLength, Npp64fc \* pDp, Npp8u \* pDeviceBuffer)**

64-bit float and 64-bit float complex dot product method, return value is 64-bit float complex.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_64f64fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.25 NppStatus nppsDotProd\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, int *nLength*, Npp64fc \* *pDp*, Npp8u \* *pDeviceBuffer*)

64-bit float complex dot product method, return value is 64-bit float complex.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDp* Pointer to the dot product result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsDotProdGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.197.1.26 NppStatus nppsDotProdGetBufferSize\_16s16sc32fc (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd\_16s16sc32fc.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

#### Returns:

NPP\_SUCCESS

### 7.197.1.27 NppStatus nppsDotProdGetBufferSize\_16s16sc32sc\_Sfs (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd\_16s16sc32sc\_Sfs.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

#### Returns:

NPP\_SUCCESS

### 7.197.1.28 NppStatus nppsDotProdGetBufferSize\_16s16sc64sc (int *nLength*, int \* *hpBufferSize*)

Device-buffer size (in bytes) for nppsDotProd\_16s16sc64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.29 NppStatus nppsDotProdGetBufferSize\_16s16sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsDotProd\_16s16sc\_Sfs*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.30 NppStatus nppsDotProdGetBufferSize\_16s32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsDotProd\_16s32f*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.31 NppStatus nppsDotProdGetBufferSize\_16s32s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for *nppsDotProd\_16s32s32s\_Sfs*.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.32 NppStatus nppsDotProdGetBufferSize\_16s32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_16s32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.33 NppStatus nppsDotProdGetBufferSize\_16s64s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_16s64s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.34 NppStatus nppsDotProdGetBufferSize\_16s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_16s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.35 NppStatus nppsDotProdGetBufferSize\_16sc32fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_16sc32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.36 NppStatus nppsDotProdGetBufferSize\_16sc32sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_16sc32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.37 NppStatus nppsDotProdGetBufferSize\_16sc64sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_16sc64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.38 NppStatus nppsDotProdGetBufferSize\_16sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_16sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.39 NppStatus nppsDotProdGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS



**7.197.1.40 NppStatus nppsDotProdGetBufferSize\_32f32fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32f32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.41 NppStatus nppsDotProdGetBufferSize\_32f32fc64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32f32fc64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.42 NppStatus nppsDotProdGetBufferSize\_32f64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32f64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.43 NppStatus nppsDotProdGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.44 NppStatus nppsDotProdGetBufferSize\_32fc64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32fc64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.45 NppStatus nppsDotProdGetBufferSize\_32s32sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32s32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.46 NppStatus nppsDotProdGetBufferSize\_32s\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32s\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.47 NppStatus nppsDotProdGetBufferSize\_32sc\_Sfs (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_32sc\_Sfs.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.48 NppStatus nppsDotProdGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.49 NppStatus nppsDotProdGetBufferSize\_64f64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_64f64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.197.1.50 NppStatus nppsDotProdGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsDotProd\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.198 Count In Range

### Functions

- [NppStatus nppsCountInRangeGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsCountInRange\_32s.*
- [NppStatus nppsCountInRange\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, int \*pCounts, [Npp32s](#) nLowerBound, [Npp32s](#) nUpperBound, [Npp8u](#) \*pDeviceBuffer)  
*Computes the number of elements whose values fall into the specified range on a 32-bit signed integer array.*

### 7.198.1 Function Documentation

#### 7.198.1.1 [NppStatus nppsCountInRange\\_32s](#) (const [Npp32s](#) \*pSrc, int nLength, int \*pCounts, [Npp32s](#) nLowerBound, [Npp32s](#) nUpperBound, [Npp8u](#) \*pDeviceBuffer)

Computes the number of elements whose values fall into the specified range on a 32-bit signed integer array.

#### Parameters:

*pSrc* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pCounts* Pointer to the number of elements.

*nLowerBound* Lower bound of the specified range.

*nUpperBound* Upper bound of the specified range.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsCountInRangeGetBufferSize\\_32s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.198.1.2 [NppStatus nppsCountInRangeGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)

Device-buffer size (in bytes) for nppsCountInRange\_32s.

#### Parameters:

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

#### Returns:

NPP\_SUCCESS

## 7.199 Count Zero Crossings

### Functions

- [NppStatus nppsZeroCrossingGetBufferSize\\_16s32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsZeroCrossing\_16s32f.*
- [NppStatus nppsZeroCrossing\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer zero crossing method, return value is 32-bit floating point.*
- [NppStatus nppsZeroCrossingGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsZeroCrossing\_32f.*
- [NppStatus nppsZeroCrossing\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating-point zero crossing method, return value is 32-bit floating point.*

### 7.199.1 Function Documentation

#### 7.199.1.1 [NppStatus nppsZeroCrossing\\_16s32f](#) (const [Npp16s](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)

16-bit signed short integer zero crossing method, return value is 32-bit floating point.

##### Parameters:

[pSrc](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pValZC](#) Pointer to the output result.

[tZCType](#) Type of the zero crossing measure: [nppZCR](#), [nppZCXor](#) or [nppZCC](#).

[pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsZeroCrossingGetBufferSize\\_16s32f](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.199.1.2 [NppStatus nppsZeroCrossing\\_32f](#) (const [Npp32f](#) \*pSrc, int nLength, [Npp32f](#) \*pValZC, [NppsZCType](#) tZCType, [Npp8u](#) \*pDeviceBuffer)

32-bit floating-point zero crossing method, return value is 32-bit floating point.

##### Parameters:

[pSrc](#) [Source Signal Pointer](#).

[nLength](#) [Signal Length](#).

[pValZC](#) Pointer to the output result.

*tZCType* Type of the zero crossing measure: nppZCR, nppZCXor or nppZCC.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsZeroCrossingGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.199.1.3 NppStatus nppsZeroCrossingGetBufferSize\_16s32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsZeroCrossing\_16s32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.199.1.4 NppStatus nppsZeroCrossingGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsZeroCrossing\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: hpBufferSize is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.200 MaximumError

Primitives for computing the maximum error between two signals.

### Functions

- [NppStatus nppsMaximumError\\_8u](#) (const [Npp8u](#) \*pSrc1, const [Npp8u](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*8-bit unsigned char maximum method.*
- [NppStatus nppsMaximumError\\_8s](#) (const [Npp8s](#) \*pSrc1, const [Npp8s](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*8-bit signed char maximum method.*
- [NppStatus nppsMaximumError\\_16u](#) (const [Npp16u](#) \*pSrc1, const [Npp16u](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*16-bit unsigned short integer maximum method.*
- [NppStatus nppsMaximumError\\_16s](#) (const [Npp16s](#) \*pSrc1, const [Npp16s](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*16-bit signed short integer maximum method.*
- [NppStatus nppsMaximumError\\_16sc](#) (const [Npp16sc](#) \*pSrc1, const [Npp16sc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*16-bit unsigned short complex integer maximum method.*
- [NppStatus nppsMaximumError\\_32u](#) (const [Npp32u](#) \*pSrc1, const [Npp32u](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit unsigned short integer maximum method.*
- [NppStatus nppsMaximumError\\_32s](#) (const [Npp32s](#) \*pSrc1, const [Npp32s](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit signed short integer maximum method.*
- [NppStatus nppsMaximumError\\_32sc](#) (const [Npp32sc](#) \*pSrc1, const [Npp32sc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit unsigned short complex integer maximum method.*
- [NppStatus nppsMaximumError\\_64s](#) (const [Npp64s](#) \*pSrc1, const [Npp64s](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit signed short integer maximum method.*
- [NppStatus nppsMaximumError\\_64sc](#) (const [Npp64sc](#) \*pSrc1, const [Npp64sc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit unsigned short complex integer maximum method.*
- [NppStatus nppsMaximumError\\_32f](#) (const [Npp32f](#) \*pSrc1, const [Npp32f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating point maximum method.*

- [NppStatus nppsMaximumError\\_32fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating point complex maximum method.*
- [NppStatus nppsMaximumError\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point maximum method.*
- [NppStatus nppsMaximumError\\_64fc](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point complex maximum method.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_8u.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_8s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_8s.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_16u.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_16s.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_16sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_16sc.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_32u.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_32s.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_32sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_32sc.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_64s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_64s.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_64sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_64sc.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_32f.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_32fc.*
- [NppStatus nppsMaximumErrorGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumError\_64f.*



- [NppStatus nppsMaximumErrorGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)

*Device-buffer size (in bytes) for nppsMaximumError\_64fc.*

### 7.200.1 Detailed Description

Primitives for computing the maximum error between two signals.

Given two signals *pSrc1* and *pSrc2* both with length *N*, the maximum error is defined as the largest absolute difference between the corresponding elements of two signals.

If the signal is in complex format, the absolute value of the complex number is used.

### 7.200.2 Function Documentation

#### 7.200.2.1 NppStatus nppsMaximumError\_16s (const Npp16s \*pSrc1, const Npp16s \*pSrc2, int nLength, Npp64f \*pDst, Npp8u \*pDeviceBuffer)

16-bit signed short integer maximum method.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_16s](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.200.2.2 NppStatus nppsMaximumError\_16sc (const Npp16sc \*pSrc1, const Npp16sc \*pSrc2, int nLength, Npp64f \*pDst, Npp8u \*pDeviceBuffer)

16-bit unsigned short complex integer maximum method.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_16sc](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.3 NppStatus nppsMaximumError\_16u (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

16-bit unsigned short integer maximum method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_16u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.4 NppStatus nppsMaximumError\_32f (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

32-bit floating point maximum method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.5 NppStatus nppsMaximumError\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

32-bit floating point complex maximum method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.6 NppStatus nppsMaximumError\_32s** (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit signed short integer maximum method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_32s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.7 NppStatus nppsMaximumError\_32sc** (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit unsigned short complex integer maximum method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_32sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.8 NppStatus nppsMaximumError\_32u** (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit unsigned short integer maximum method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_32u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.200.2.9 **NppStatus nppsMaximumError\_64f** (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

64-bit floating point maximum method.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.200.2.10 **NppStatus nppsMaximumError\_64fc** (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

64-bit floating point complex maximum method.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.200.2.11 **NppStatus nppsMaximumError\_64s** (const Npp64s \* *pSrc1*, const Npp64s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

64-bit signed short integer maximum method.

#### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_64s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.12 NppStatus nppsMaximumError\_64sc (const Npp64sc \* *pSrc1*, const Npp64sc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

64-bit unsigned short complex integer maximum method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_64sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.13 NppStatus nppsMaximumError\_8s (const Npp8s \* *pSrc1*, const Npp8s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

8-bit signed char maximum method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_8s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.14 NppStatus nppsMaximumError\_8u (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

8-bit unsigned char maximum method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumErrorGetBufferSize\\_8u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.200.2.15 NppStatus nppsMaximumErrorGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.16 NppStatus nppsMaximumErrorGetBufferSize\_16sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_16sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.17 NppStatus nppsMaximumErrorGetBufferSize\_16u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.18 NppStatus nppsMaximumErrorGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.19 NppStatus nppsMaximumErrorGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.20 NppStatus nppsMaximumErrorGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.21 NppStatus nppsMaximumErrorGetBufferSize\_32sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_32sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.22 NppStatus nppsMaximumErrorGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_32u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.23 NppStatus nppsMaximumErrorGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.24 NppStatus nppsMaximumErrorGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.25 NppStatus nppsMaximumErrorGetBufferSize\_64s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_64s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.26 NppStatus nppsMaximumErrorGetBufferSize\_64sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS



**7.200.2.27 NppStatus nppsMaximumErrorGetBufferSize\_8s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_8s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.200.2.28 NppStatus nppsMaximumErrorGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumError\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.201 AverageError

Primitives for computing the Average error between two signals.

### Functions

- `NppStatus nppsAverageError_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*8-bit unsigned char Average method.*
- `NppStatus nppsAverageError_8s` (const `Npp8s` \*pSrc1, const `Npp8s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*8-bit signed char Average method.*
- `NppStatus nppsAverageError_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit unsigned short integer Average method.*
- `NppStatus nppsAverageError_16s` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit signed short integer Average method.*
- `NppStatus nppsAverageError_16sc` (const `Npp16sc` \*pSrc1, const `Npp16sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit unsigned short complex integer Average method.*
- `NppStatus nppsAverageError_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit unsigned short integer Average method.*
- `NppStatus nppsAverageError_32s` (const `Npp32s` \*pSrc1, const `Npp32s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit signed short integer Average method.*
- `NppStatus nppsAverageError_32sc` (const `Npp32sc` \*pSrc1, const `Npp32sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit unsigned short complex integer Average method.*
- `NppStatus nppsAverageError_64s` (const `Npp64s` \*pSrc1, const `Npp64s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*64-bit signed short integer Average method.*
- `NppStatus nppsAverageError_64sc` (const `Npp64sc` \*pSrc1, const `Npp64sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*64-bit unsigned short complex integer Average method.*
- `NppStatus nppsAverageError_32f` (const `Npp32f` \*pSrc1, const `Npp32f` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit floating point Average method.*

- [NppStatus nppsAverageError\\_32fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating point complex Average method.*
- [NppStatus nppsAverageError\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point Average method.*
- [NppStatus nppsAverageError\\_64fc](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point complex Average method.*
- [NppStatus nppsAverageErrorGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_8u.*
- [NppStatus nppsAverageErrorGetBufferSize\\_8s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_8s.*
- [NppStatus nppsAverageErrorGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_16u.*
- [NppStatus nppsAverageErrorGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_16s.*
- [NppStatus nppsAverageErrorGetBufferSize\\_16sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_16sc.*
- [NppStatus nppsAverageErrorGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_32u.*
- [NppStatus nppsAverageErrorGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_32s.*
- [NppStatus nppsAverageErrorGetBufferSize\\_32sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_32sc.*
- [NppStatus nppsAverageErrorGetBufferSize\\_64s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_64s.*
- [NppStatus nppsAverageErrorGetBufferSize\\_64sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_64sc.*
- [NppStatus nppsAverageErrorGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_32f.*
- [NppStatus nppsAverageErrorGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_32fc.*
- [NppStatus nppsAverageErrorGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageError\_64f.*

- [NppStatus nppsAverageErrorGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for [nppsAverageError\\_64fc](#).

### 7.201.1 Detailed Description

Primitives for computing the Average error between two signals.

Given two signals  $pSrc1$  and  $pSrc2$  both with length  $N$ , the average error is defined as

$$AverageError = \frac{1}{N} \sum_{n=0}^{N-1} |pSrc1(n) - pSrc2(n)|$$

If the signal is in complex format, the absolute value of the complex number is used.

### 7.201.2 Function Documentation

#### 7.201.2.1 NppStatus nppsAverageError\_16s (const Npp16s \* pSrc1, const Npp16s \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)

16-bit signed short integer Average method.

##### Parameters:

- [pSrc1](#) Source Signal Pointer.
- [pSrc2](#) Source Signal Pointer.
- [nLength](#) Signal Length.
- [pDst](#) Pointer to the error result.
- [pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_16s](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.201.2.2 NppStatus nppsAverageError\_16sc (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)

16-bit unsigned short complex integer Average method.

##### Parameters:

- [pSrc1](#) Source Signal Pointer.
- [pSrc2](#) Source Signal Pointer.
- [nLength](#) Signal Length.
- [pDst](#) Pointer to the error result.
- [pDeviceBuffer](#) Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_16sc](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.201.2.3 NppStatus nppsAverageError\_16u (const Npp16u \* *pSrc1*, const Npp16u \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

16-bit unsigned short integer Average method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_16u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.201.2.4 NppStatus nppsAverageError\_32f (const Npp32f \* *pSrc1*, const Npp32f \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

32-bit floating point Average method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.201.2.5 NppStatus nppsAverageError\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)**

32-bit floating point complex Average method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.6 **NppStatus nppsAverageError\_32s** (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit signed short integer Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_32s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.7 **NppStatus nppsAverageError\_32sc** (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit unsigned short complex integer Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_32sc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.8 **NppStatus nppsAverageError\_32u** (const Npp32u \* *pSrc1*, const Npp32u \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit unsigned short integer Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_32u](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.9 NppStatus nppsAverageError\_64f (const Npp64f \* *pSrc1*, const Npp64f \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

64-bit floating point Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_64f](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.10 NppStatus nppsAverageError\_64fc (const Npp64fc \* *pSrc1*, const Npp64fc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

64-bit floating point complex Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.11 NppStatus nppsAverageError\_64s (const Npp64s \* *pSrc1*, const Npp64s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

64-bit signed short integer Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_64s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.12 **NppStatus nppsAverageError\_64sc** (const Npp64sc \* *pSrc1*, const Npp64sc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

64-bit unsigned short complex integer Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_64sc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.13 **NppStatus nppsAverageError\_8s** (const Npp8s \* *pSrc1*, const Npp8s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

8-bit signed char Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_8s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.201.2.14 **NppStatus nppsAverageError\_8u** (const Npp8u \* *pSrc1*, const Npp8u \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

8-bit unsigned char Average method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageErrorGetBufferSize\\_8u](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.201.2.15 NppStatus nppsAverageErrorGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.16 NppStatus nppsAverageErrorGetBufferSize\_16sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_16sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.17 NppStatus nppsAverageErrorGetBufferSize\_16u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.18 NppStatus nppsAverageErrorGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.19 NppStatus nppsAverageErrorGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.20 NppStatus nppsAverageErrorGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.21 NppStatus nppsAverageErrorGetBufferSize\_32sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_32sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.22 NppStatus nppsAverageErrorGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_32u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.23 NppStatus nppsAverageErrorGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.24 NppStatus nppsAverageErrorGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.25 NppStatus nppsAverageErrorGetBufferSize\_64s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_64s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.26 NppStatus nppsAverageErrorGetBufferSize\_64sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.27 NppStatus nppsAverageErrorGetBufferSize\_8s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_8s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.201.2.28 NppStatus nppsAverageErrorGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageError\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.202 MaximumRelativeError

Primitives for computing the MaximumRelative error between two signals.

### Functions

- `NppStatus nppsMaximumRelativeError_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*8-bit unsigned char MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_8s` (const `Npp8s` \*pSrc1, const `Npp8s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*8-bit signed char MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit unsigned short integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_16s` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit signed short integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_16sc` (const `Npp16sc` \*pSrc1, const `Npp16sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit unsigned short complex integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit unsigned short integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_32s` (const `Npp32s` \*pSrc1, const `Npp32s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit signed short integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_32sc` (const `Npp32sc` \*pSrc1, const `Npp32sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit unsigned short complex integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_64s` (const `Npp64s` \*pSrc1, const `Npp64s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*64-bit signed short integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_64sc` (const `Npp64sc` \*pSrc1, const `Npp64sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*64-bit unsigned short complex integer MaximumRelative method.*
- `NppStatus nppsMaximumRelativeError_32f` (const `Npp32f` \*pSrc1, const `Npp32f` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit floating point MaximumRelative method.*

- [NppStatus nppsMaximumRelativeError\\_32fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating point complex MaximumRelative method.*
- [NppStatus nppsMaximumRelativeError\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point MaximumRelative method.*
- [NppStatus nppsMaximumRelativeError\\_64fc](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point complex MaximumRelative method.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_8u.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_8s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_8s.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_16u.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_16s.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_16sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_16sc.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_32u.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_32s.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_32sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_32sc.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_64s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_64s.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_64sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_64sc.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_32f.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_32fc.*
- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsMaximumRelativeError\_64f.*

- [NppStatus nppsMaximumRelativeErrorGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for *nppsMaximumRelativeError\_64fc*.

### 7.202.1 Detailed Description

Primitives for computing the MaximumRelative error between two signals.

Given two signals *pSrc1* and *pSrc2* both with length *N*, the maximum relative error is defined as

$$\text{MaximumRelativeError} = \max \frac{|pSrc1(n) - pSrc2(n)|}{\max(|pSrc1(n)|, |pSrc2(n)|)}$$

If the signal is in complex format, the absolute value of the complex number is used.

### 7.202.2 Function Documentation

#### 7.202.2.1 NppStatus nppsMaximumRelativeError\_16s (const Npp16s \* pSrc1, const Npp16s \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)

16-bit signed short integer MaximumRelative method.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_16s](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.202.2.2 NppStatus nppsMaximumRelativeError\_16sc (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)

16-bit unsigned short complex integer MaximumRelative method.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_16sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.3 NppStatus nppsMaximumRelativeError\_16u (const Npp16u \* pSrc1, const Npp16u \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

16-bit unsigned short integer MaximumRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_16u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.4 NppStatus nppsMaximumRelativeError\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

32-bit floating point MaximumRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



### 7.202.2.5 NppStatus nppsMaximumRelativeError\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit floating point complex MaximumRelative method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.202.2.6 NppStatus nppsMaximumRelativeError\_32s (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit signed short integer MaximumRelative method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_32s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.202.2.7 NppStatus nppsMaximumRelativeError\_32sc (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit unsigned short complex integer MaximumRelative method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_32sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.8** `NppStatus nppsMaximumRelativeError_32u (const Npp32u * pSrc1, const Npp32u * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

32-bit unsigned short integer MaximumRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_32u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.9** `NppStatus nppsMaximumRelativeError_64f (const Npp64f * pSrc1, const Npp64f * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit floating point MaximumRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.10** `NppStatus nppsMaximumRelativeError_64fc (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit floating point complex MaximumRelative method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.11** `NppStatus nppsMaximumRelativeError_64s (const Npp64s * pSrc1, const Npp64s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit signed short integer MaximumRelative method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_64s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.12** `NppStatus nppsMaximumRelativeError_64sc (const Npp64sc * pSrc1, const Npp64sc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit unsigned short complex integer MaximumRelative method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_64sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.13 NppStatus nppsMaximumRelativeError\_8s (const Npp8s \* pSrc1, const Npp8s \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

8-bit signed char MaximumRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_8s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.14 NppStatus nppsMaximumRelativeError\_8u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

8-bit unsigned char MaximumRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsMaximumRelativeErrorGetBufferSize\\_8u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.202.2.15 NppStatus nppsMaximumRelativeErrorGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.16 NppStatus nppsMaximumRelativeErrorGetBufferSize\_16sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_16sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.17 NppStatus nppsMaximumRelativeErrorGetBufferSize\_16u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.18 NppStatus nppsMaximumRelativeErrorGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.19 NppStatus nppsMaximumRelativeErrorGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.20 NppStatus nppsMaximumRelativeErrorGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.21 NppStatus nppsMaximumRelativeErrorGetBufferSize\_32sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_32sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.22 NppStatus nppsMaximumRelativeErrorGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_32u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.23 NppStatus nppsMaximumRelativeErrorGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.24 NppStatus nppsMaximumRelativeErrorGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.25 NppStatus nppsMaximumRelativeErrorGetBufferSize\_64s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_64s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.26 NppStatus nppsMaximumRelativeErrorGetBufferSize\_64sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.27 NppStatus nppsMaximumRelativeErrorGetBufferSize\_8s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_8s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.202.2.28 NppStatus nppsMaximumRelativeErrorGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsMaximumRelativeError\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS



## 7.203 AverageRelativeError

Primitives for computing the AverageRelative error between two signals.

### Functions

- `NppStatus nppsAverageRelativeError_8u` (const `Npp8u` \*pSrc1, const `Npp8u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*8-bit unsigned char AverageRelative method.*
- `NppStatus nppsAverageRelativeError_8s` (const `Npp8s` \*pSrc1, const `Npp8s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*8-bit signed char AverageRelative method.*
- `NppStatus nppsAverageRelativeError_16u` (const `Npp16u` \*pSrc1, const `Npp16u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit unsigned short integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_16s` (const `Npp16s` \*pSrc1, const `Npp16s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit signed short integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_16sc` (const `Npp16sc` \*pSrc1, const `Npp16sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*16-bit unsigned short complex integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_32u` (const `Npp32u` \*pSrc1, const `Npp32u` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit unsigned short integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_32s` (const `Npp32s` \*pSrc1, const `Npp32s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit signed short integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_32sc` (const `Npp32sc` \*pSrc1, const `Npp32sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit unsigned short complex integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_64s` (const `Npp64s` \*pSrc1, const `Npp64s` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*64-bit signed short integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_64sc` (const `Npp64sc` \*pSrc1, const `Npp64sc` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*64-bit unsigned short complex integer AverageRelative method.*
- `NppStatus nppsAverageRelativeError_32f` (const `Npp32f` \*pSrc1, const `Npp32f` \*pSrc2, int nLength, `Npp64f` \*pDst, `Npp8u` \*pDeviceBuffer)  
*32-bit floating point AverageRelative method.*

- [NppStatus nppsAverageRelativeError\\_32fc](#) (const [Npp32fc](#) \*pSrc1, const [Npp32fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*32-bit floating point complex AverageRelative method.*
- [NppStatus nppsAverageRelativeError\\_64f](#) (const [Npp64f](#) \*pSrc1, const [Npp64f](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point AverageRelative method.*
- [NppStatus nppsAverageRelativeError\\_64fc](#) (const [Npp64fc](#) \*pSrc1, const [Npp64fc](#) \*pSrc2, int nLength, [Npp64f](#) \*pDst, [Npp8u](#) \*pDeviceBuffer)  
*64-bit floating point complex AverageRelative method.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_8u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_8u.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_8s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_8s.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_16u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_16u.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_16s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_16s.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_16sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_16sc.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_32u](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_32u.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_32s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_32s.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_32sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_32sc.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_64s](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_64s.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_64sc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_64sc.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_32f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_32f.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_32fc](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_32fc.*
- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_64f](#) (int nLength, int \*hpBufferSize)  
*Device-buffer size (in bytes) for nppsAverageRelativeError\_64f.*

- [NppStatus nppsAverageRelativeErrorGetBufferSize\\_64fc](#) (int nLength, int \*hpBufferSize)  
Device-buffer size (in bytes) for [nppsAverageRelativeError\\_64fc](#).

### 7.203.1 Detailed Description

Primitives for computing the AverageRelative error between two signals.

Given two signals  $pSrc1$  and  $pSrc2$  both with length  $N$ , the average relative error is defined as

$$AverageRelativeError = \frac{1}{N} \sum_{n=0}^{N-1} \frac{|pSrc1(n) - pSrc2(n)|}{\max(|pSrc1(n)|, |pSrc2(n)|)}$$

If the signal is in complex format, the absolute value of the complex number is used.

### 7.203.2 Function Documentation

#### 7.203.2.1 NppStatus nppsAverageRelativeError\_16s (const Npp16s \* pSrc1, const Npp16s \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)

16-bit signed short integer AverageRelative method.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_16s](#) to determine the minium number of bytes required.

##### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

#### 7.203.2.2 NppStatus nppsAverageRelativeError\_16sc (const Npp16sc \* pSrc1, const Npp16sc \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)

16-bit unsigned short complex integer AverageRelative method.

##### Parameters:

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_16sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.3 NppStatus nppsAverageRelativeError\_16u (const Npp16u \* pSrc1, const Npp16u \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

16-bit unsigned short integer AverageRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_16u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.4 NppStatus nppsAverageRelativeError\_32f (const Npp32f \* pSrc1, const Npp32f \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

32-bit floating point AverageRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_32f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.203.2.5 NppStatus nppsAverageRelativeError\_32fc (const Npp32fc \* *pSrc1*, const Npp32fc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit floating point complex AverageRelative method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_32fc](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.203.2.6 NppStatus nppsAverageRelativeError\_32s (const Npp32s \* *pSrc1*, const Npp32s \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit signed short integer AverageRelative method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_32s](#) to determine the minium number of bytes required.

#### Returns:

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

### 7.203.2.7 NppStatus nppsAverageRelativeError\_32sc (const Npp32sc \* *pSrc1*, const Npp32sc \* *pSrc2*, int *nLength*, Npp64f \* *pDst*, Npp8u \* *pDeviceBuffer*)

32-bit unsigned short complex integer AverageRelative method.

#### Parameters:

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_32sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.8 NppStatus nppsAverageRelativeError\_32u (const Npp32u \* pSrc1, const Npp32u \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

32-bit unsigned short integer AverageRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_32u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.9 NppStatus nppsAverageRelativeError\_64f (const Npp64f \* pSrc1, const Npp64f \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

64-bit floating point AverageRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_64f](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.10** `NppStatus nppsAverageRelativeError_64fc (const Npp64fc * pSrc1, const Npp64fc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit floating point complex AverageRelative method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_64fc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.11** `NppStatus nppsAverageRelativeError_64s (const Npp64s * pSrc1, const Npp64s * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit signed short integer AverageRelative method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_64s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.12** `NppStatus nppsAverageRelativeError_64sc (const Npp64sc * pSrc1, const Npp64sc * pSrc2, int nLength, Npp64f * pDst, Npp8u * pDeviceBuffer)`

64-bit unsigned short complex integer AverageRelative method.

**Parameters:**

*pSrc1* Source Signal Pointer.

*pSrc2* Source Signal Pointer.

*nLength* Signal Length.

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_64sc](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.13 NppStatus nppsAverageRelativeError\_8s (const Npp8s \* pSrc1, const Npp8s \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

8-bit signed char AverageRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_8s](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).

**7.203.2.14 NppStatus nppsAverageRelativeError\_8u (const Npp8u \* pSrc1, const Npp8u \* pSrc2, int nLength, Npp64f \* pDst, Npp8u \* pDeviceBuffer)**

8-bit unsigned char AverageRelative method.

**Parameters:**

*pSrc1* [Source Signal Pointer](#).

*pSrc2* [Source Signal Pointer](#).

*nLength* [Signal Length](#).

*pDst* Pointer to the error result.

*pDeviceBuffer* Pointer to the required device memory allocation, [Scratch Buffer and Host Pointer](#).  
Use [nppsAverageRelativeErrorGetBufferSize\\_8u](#) to determine the minium number of bytes required.

**Returns:**

[Signal Data Related Error Codes](#), [Length Related Error Codes](#).



**7.203.2.15 NppStatus nppsAverageRelativeErrorGetBufferSize\_16s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_16s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.16 NppStatus nppsAverageRelativeErrorGetBufferSize\_16sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_16sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.17 NppStatus nppsAverageRelativeErrorGetBufferSize\_16u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_16u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.18 NppStatus nppsAverageRelativeErrorGetBufferSize\_32f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_32f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.19 NppStatus nppsAverageRelativeErrorGetBufferSize\_32fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_32fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.20 NppStatus nppsAverageRelativeErrorGetBufferSize\_32s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_32s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.21 NppStatus nppsAverageRelativeErrorGetBufferSize\_32sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_32sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.22 NppStatus nppsAverageRelativeErrorGetBufferSize\_32u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_32u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.23 NppStatus nppsAverageRelativeErrorGetBufferSize\_64f (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_64f.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.24 NppStatus nppsAverageRelativeErrorGetBufferSize\_64fc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_64fc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.25 NppStatus nppsAverageRelativeErrorGetBufferSize\_64s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_64s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.26 NppStatus nppsAverageRelativeErrorGetBufferSize\_64sc (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_64sc.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.27 NppStatus nppsAverageRelativeErrorGetBufferSize\_8s (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_8s.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

**7.203.2.28 NppStatus nppsAverageRelativeErrorGetBufferSize\_8u (int *nLength*, int \* *hpBufferSize*)**

Device-buffer size (in bytes) for nppsAverageRelativeError\_8u.

**Parameters:**

*nLength* [Signal Length](#).

*hpBufferSize* Required buffer size. Important: *hpBufferSize* is a *host pointer*.

**Returns:**

NPP\_SUCCESS

## 7.204 Memory Management

### Modules

- [Malloc](#)

*Signal-allocator methods for allocating 1D arrays of data in device memory.*

- [Free](#)

*Free signal memory.*

## 7.205 Malloc

Signal-allocator methods for allocating 1D arrays of data in device memory.

### Functions

- [Npp8u](#) \* [nppsMalloc\\_8u](#) (int nSize)  
*8-bit unsigned signal allocator.*
- [Npp8s](#) \* [nppsMalloc\\_8s](#) (int nSize)  
*8-bit signed signal allocator.*
- [Npp16u](#) \* [nppsMalloc\\_16u](#) (int nSize)  
*16-bit unsigned signal allocator.*
- [Npp16s](#) \* [nppsMalloc\\_16s](#) (int nSize)  
*16-bit signal allocator.*
- [Npp16sc](#) \* [nppsMalloc\\_16sc](#) (int nSize)  
*16-bit complex-value signal allocator.*
- [Npp32u](#) \* [nppsMalloc\\_32u](#) (int nSize)  
*32-bit unsigned signal allocator.*
- [Npp32s](#) \* [nppsMalloc\\_32s](#) (int nSize)  
*32-bit integer signal allocator.*
- [Npp32sc](#) \* [nppsMalloc\\_32sc](#) (int nSize)  
*32-bit complex integer signal allocator.*
- [Npp32f](#) \* [nppsMalloc\\_32f](#) (int nSize)  
*32-bit float signal allocator.*
- [Npp32fc](#) \* [nppsMalloc\\_32fc](#) (int nSize)  
*32-bit complex float signal allocator.*
- [Npp64s](#) \* [nppsMalloc\\_64s](#) (int nSize)  
*64-bit long integer signal allocator.*
- [Npp64sc](#) \* [nppsMalloc\\_64sc](#) (int nSize)  
*64-bit complex long integer signal allocator.*
- [Npp64f](#) \* [nppsMalloc\\_64f](#) (int nSize)  
*64-bit float (double) signal allocator.*
- [Npp64fc](#) \* [nppsMalloc\\_64fc](#) (int nSize)  
*64-bit complex complex signal allocator.*

### 7.205.1 Detailed Description

Signal-allocator methods for allocating 1D arrays of data in device memory.

All allocators have size parameters to specify the size of the signal (1D array) being allocated.

The allocator methods return a pointer to the newly allocated memory of appropriate type. If device-memory allocation is not possible due to resource constraints the allocators return 0 (i.e. NULL pointer).

All signal allocators allocate memory aligned such that it is beneficial to the performance of the majority of the signal-processing primitives. It is no mandatory however to use these allocators. Any valid CUDA device-memory pointers can be passed to NPP primitives.

### 7.205.2 Function Documentation

#### 7.205.2.1 Npp16s\* nppsMalloc\_16s (int *nSize*)

16-bit signal allocator.

**Parameters:**

*nSize* Number of shorts in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

#### 7.205.2.2 Npp16sc\* nppsMalloc\_16sc (int *nSize*)

16-bit complex-value signal allocator.

**Parameters:**

*nSize* Number of 16-bit complex numbers in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

#### 7.205.2.3 Npp16u\* nppsMalloc\_16u (int *nSize*)

16-bit unsigned signal allocator.

**Parameters:**

*nSize* Number of unsigned shorts in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.4 Npp32f\* nppsMalloc\_32f (int *nSize*)**

32-bit float signal allocator.

**Parameters:**

*nSize* Number of floats in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.5 Npp32fc\* nppsMalloc\_32fc (int *nSize*)**

32-bit complex float signal allocator.

**Parameters:**

*nSize* Number of complex float values in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.6 Npp32s\* nppsMalloc\_32s (int *nSize*)**

32-bit integer signal allocator.

**Parameters:**

*nSize* Number of ints in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.7 Npp32sc\* nppsMalloc\_32sc (int *nSize*)**

32-bit complex integer signal allocator.

**Parameters:**

*nSize* Number of complex integner values in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.



**7.205.2.8 Npp32u\* nppsMalloc\_32u (int nSize)**

32-bit unsigned signal allocator.

**Parameters:**

*nSize* Number of unsigned ints in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.9 Npp64f\* nppsMalloc\_64f (int nSize)**

64-bit float (double) signal allocator.

**Parameters:**

*nSize* Number of doubles in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.10 Npp64fc\* nppsMalloc\_64fc (int nSize)**

64-bit complex complex signal allocator.

**Parameters:**

*nSize* Number of complex double values in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.11 Npp64s\* nppsMalloc\_64s (int nSize)**

64-bit long integer signal allocator.

**Parameters:**

*nSize* Number of long ints in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.12 Npp64sc\* nppsMalloc\_64sc (int *nSize*)**

64-bit complex long integer signal allocator.

**Parameters:**

*nSize* Number of complex long int values in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.13 Npp8s\* nppsMalloc\_8s (int *nSize*)**

8-bit signed signal allocator.

**Parameters:**

*nSize* Number of (signed) chars in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

**7.205.2.14 Npp8u\* nppsMalloc\_8u (int *nSize*)**

8-bit unsigned signal allocator.

**Parameters:**

*nSize* Number of unsigned chars in the new signal.

**Returns:**

A pointer to the new signal. 0 (NULL-pointer) indicates that an error occurred during allocation.

## 7.206 Free

Free signal memory.

### Functions

- void `nppsFree` (void \**pValues*)  
*Free method for any signal memory.*

### 7.206.1 Detailed Description

Free signal memory.

### 7.206.2 Function Documentation

#### 7.206.2.1 void `nppsFree` (void \* *pValues*)

Free method for any signal memory.

#### Parameters:

*pValues* A pointer to memory allocated using `nppiMalloc_<modifier>`.



# Chapter 8

## Data Structure Documentation

### 8.1 NPP\_ALIGN\_16 Struct Reference

Complex Number This struct represents a long long complex number.

```
#include <nppdefs.h>
```

#### Data Fields

- [Npp64s re](#)  
*Real part.*
- [Npp64s im](#)  
*Imaginary part.*
- [Npp64f re](#)  
*Real part.*
- [Npp64f im](#)  
*Imaginary part.*

#### 8.1.1 Detailed Description

Complex Number This struct represents a long long complex number.

Complex Number This struct represents a double floating-point complex number.

#### 8.1.2 Field Documentation

##### 8.1.2.1 Npp64f NPP\_ALIGN\_16::im

Imaginary part.

**8.1.2.2 Npp64s NPP\_ALIGN\_16::im**

Imaginary part.

**8.1.2.3 Npp64f NPP\_ALIGN\_16::re**

Real part.

**8.1.2.4 Npp64s NPP\_ALIGN\_16::re**

Real part.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h

## 8.2 NPP\_ALIGN\_8 Struct Reference

Complex Number This struct represents an unsigned int complex number.

```
#include <nppdefs.h>
```

### Data Fields

- [Npp32u re](#)  
*Real part.*
- [Npp32u im](#)  
*Imaginary part.*
- [Npp32s re](#)  
*Real part.*
- [Npp32s im](#)  
*Imaginary part.*
- [Npp32f re](#)  
*Real part.*
- [Npp32f im](#)  
*Imaginary part.*

### 8.2.1 Detailed Description

Complex Number This struct represents an unsigned int complex number.

Complex Number This struct represents a single floating-point complex number.

Complex Number This struct represents a signed int complex number.

### 8.2.2 Field Documentation

#### 8.2.2.1 Npp32f NPP\_ALIGN\_8::im

Imaginary part.

#### 8.2.2.2 Npp32s NPP\_ALIGN\_8::im

Imaginary part.

#### 8.2.2.3 Npp32u NPP\_ALIGN\_8::im

Imaginary part.

**8.2.2.4 Npp32f NPP\_ALIGN\_8::re**

Real part.

**8.2.2.5 Npp32s NPP\_ALIGN\_8::re**

Real part.

**8.2.2.6 Npp32u NPP\_ALIGN\_8::re**

Real part.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h



## 8.3 NppiHaarBuffer Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int [haarBufferSize](#)  
*size of the buffer*
- Npp32s \* [haarBuffer](#)  
*buffer*

### 8.3.1 Field Documentation

#### 8.3.1.1 Npp32s\* NppiHaarBuffer::haarBuffer

buffer

#### 8.3.1.2 int NppiHaarBuffer::haarBufferSize

size of the buffer

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h

## 8.4 NppiHaarClassifier\_32f Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int [numClassifiers](#)  
*number of classifiers*
- [Npp32s](#) \* [classifiers](#)  
*packed classifier data 40 bytes each*
- [size\\_t](#) [classifierStep](#)
- [NppiSize](#) [classifierSize](#)
- [Npp32s](#) \* [counterDevice](#)

### 8.4.1 Field Documentation

#### 8.4.1.1 [Npp32s](#)\* [NppiHaarClassifier\\_32f::classifiers](#)

packed classifier data 40 bytes each

#### 8.4.1.2 [NppiSize](#) [NppiHaarClassifier\\_32f::classifierSize](#)

#### 8.4.1.3 [size\\_t](#) [NppiHaarClassifier\\_32f::classifierStep](#)

#### 8.4.1.4 [Npp32s](#)\* [NppiHaarClassifier\\_32f::counterDevice](#)

#### 8.4.1.5 [int](#) [NppiHaarClassifier\\_32f::numClassifiers](#)

number of classifiers

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h

## 8.5 NppiPoint Struct Reference

2D Point

```
#include <nppdefs.h>
```

### Data Fields

- `int x`  
*x-coordinate.*
- `int y`  
*y-coordinate.*

### 8.5.1 Detailed Description

2D Point

### 8.5.2 Field Documentation

#### 8.5.2.1 `int NppiPoint::x`

x-coordinate.

#### 8.5.2.2 `int NppiPoint::y`

y-coordinate.

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h`

## 8.6 NppiRect Struct Reference

2D Rectangle This struct contains position and size information of a rectangle in two space.

```
#include <nppdefs.h>
```

### Data Fields

- `int x`  
*x-coordinate of upper left corner.*
- `int y`  
*y-coordinate of upper left corner.*
- `int width`  
*Rectangle width.*
- `int height`  
*Rectangle height.*

### 8.6.1 Detailed Description

2D Rectangle This struct contains position and size information of a rectangle in two space.

The rectangle's position is usually signified by the coordinate of its upper-left corner.

### 8.6.2 Field Documentation

#### 8.6.2.1 `int NppiRect::height`

Rectangle height.

#### 8.6.2.2 `int NppiRect::width`

Rectangle width.

#### 8.6.2.3 `int NppiRect::x`

x-coordinate of upper left corner.

#### 8.6.2.4 `int NppiRect::y`

y-coordinate of upper left corner.

The documentation for this struct was generated from the following file:

- `C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h`

## 8.7 NppiSize Struct Reference

2D Size This struct typically represents the size of a rectangular region in two space.

```
#include <nppdefs.h>
```

### Data Fields

- int `width`  
*Rectangle width.*
- int `height`  
*Rectangle height.*

### 8.7.1 Detailed Description

2D Size This struct typically represents the size of a rectangular region in two space.

### 8.7.2 Field Documentation

#### 8.7.2.1 int NppiSize::height

Rectangle height.

#### 8.7.2.2 int NppiSize::width

Rectangle width.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h

## 8.8 NppLibraryVersion Struct Reference

```
#include <nppdefs.h>
```

### Data Fields

- int [major](#)  
*Major version number.*
- int [minor](#)  
*Minor version number.*
- int [build](#)  
*Build number.*

### 8.8.1 Field Documentation

#### 8.8.1.1 int NppLibraryVersion::build

Build number.

This reflects the nightly build this release was made from.

#### 8.8.1.2 int NppLibraryVersion::major

Major version number.

#### 8.8.1.3 int NppLibraryVersion::minor

Minor version number.

The documentation for this struct was generated from the following file:

- C:/src/sw/rel/gpgpu/toolkit/r6.0/NPP/npp/include/nppdefs.h

# Index

- [\\_\\_align\\_\\_](#)
  - [npp\\_basic\\_types](#), [49](#), [50](#)
- [10Log10](#), [2370](#)
- [1D Linear Filter](#), [989](#)
- [1D Window Sum](#), [1046](#)
- [2D Fixed Linear Filters](#), [1112](#)
- [Abs](#), [320](#), [2344](#)
- [AbsDiff](#), [327](#)
- [AbsDiffC](#), [166](#)
- [Add](#), [168](#), [2294](#)
- [AddC](#), [55](#), [2247](#)
- [AddProduct](#), [200](#), [2306](#)
- [AddProductC](#), [2256](#)
- [AddSquare](#), [197](#)
- [AddWeighted](#), [204](#)
- [Affine Transforms](#), [1267](#)
- [Alpha Composition](#), [472](#)
- [AlphaComp](#), [488](#)
- [AlphaCompC](#), [473](#)
- [AlphaPremul](#), [495](#)
- [AlphaPremulC](#), [481](#)
- [And](#), [432](#), [2386](#)
- [AndC](#), [371](#), [2383](#)
- [Arctan](#), [2375](#)
- [Arithmetic and Logical Operations](#), [52](#), [2244](#)
- [Arithmetic Operations](#), [53](#), [2245](#)
- [AverageError](#), [2047](#), [2582](#)
- [AverageRelativeError](#), [2094](#), [2605](#)
- [Basic NPP Data Types](#), [47](#)
- [build](#)
  - [NppLibraryVersion](#), [2634](#)
- [Cauchy, CauchyD, and CauchyDD2](#), [2380](#)
- [classifiers](#)
  - [NppiHaarClassifier\\_32f](#), [2630](#)
- [classifierSize](#)
  - [NppiHaarClassifier\\_32f](#), [2630](#)
- [classifierStep](#)
  - [NppiHaarClassifier\\_32f](#), [2630](#)
- [Color and Sampling Conversion](#), [497](#)
- [Color Gamma Correction](#), [606](#)
- [Color Model Conversion](#), [498](#)
- [Color Processing](#), [615](#)
- [Color Sampling Format Conversion](#), [578](#)
- [Compare Operations](#), [2220](#)
- [Complement Color Key](#), [612](#)
- [Compression](#), [701](#)
- [Conversion Functions](#), [2412](#)
- [Convert](#), [801](#), [2413](#)
- [Convolution](#), [1057](#)
- [Copy](#), [754](#), [2453](#)
- [Copy Constant Border](#), [860](#)
- [Copy Replicate Border](#), [873](#)
- [Copy Sub-Pixel](#), [898](#)
- [Copy Wrap Border](#), [885](#)
- [core\\_npp](#)
  - [nppGetGpuComputeCapability](#), [32](#)
  - [nppGetGpuName](#), [32](#)
  - [nppGetGpuNumSMs](#), [32](#)
  - [nppGetLibVersion](#), [32](#)
  - [nppGetMaxThreadsPerBlock](#), [32](#)
  - [nppGetMaxThreadsPerSM](#), [32](#)
  - [nppGetStream](#), [33](#)
  - [nppSetStream](#), [33](#)
- [Count In Range](#), [2568](#)
- [Count Zero Crossings](#), [2569](#)
- [counterDevice](#)
  - [NppiHaarClassifier\\_32f](#), [2630](#)
- [CountInRange.](#), [1826](#)
- [CrossCorrFull\\_Norm](#), [1919](#)
- [CrossCorrFull\\_NormLevel](#), [1955](#)
- [crosscorrfullnorm](#)
  - [nppiCrossCorrFull\\_Norm\\_16u32f\\_AC4R](#), [1921](#)
  - [nppiCrossCorrFull\\_Norm\\_16u32f\\_C1R](#), [1921](#)
  - [nppiCrossCorrFull\\_Norm\\_16u32f\\_C3R](#), [1921](#)
  - [nppiCrossCorrFull\\_Norm\\_16u32f\\_C4R](#), [1922](#)
  - [nppiCrossCorrFull\\_Norm\\_32f\\_AC4R](#), [1922](#)
  - [nppiCrossCorrFull\\_Norm\\_32f\\_C1R](#), [1923](#)
  - [nppiCrossCorrFull\\_Norm\\_32f\\_C3R](#), [1923](#)
  - [nppiCrossCorrFull\\_Norm\\_32f\\_C4R](#), [1924](#)
  - [nppiCrossCorrFull\\_Norm\\_8s32f\\_AC4R](#), [1924](#)
  - [nppiCrossCorrFull\\_Norm\\_8s32f\\_C1R](#), [1924](#)
  - [nppiCrossCorrFull\\_Norm\\_8s32f\\_C3R](#), [1925](#)
  - [nppiCrossCorrFull\\_Norm\\_8s32f\\_C4R](#), [1925](#)
  - [nppiCrossCorrFull\\_Norm\\_8u32f\\_AC4R](#), [1926](#)
  - [nppiCrossCorrFull\\_Norm\\_8u32f\\_C1R](#), [1926](#)
  - [nppiCrossCorrFull\\_Norm\\_8u32f\\_C3R](#), [1927](#)

- nppiCrossCorrFull\_Norm\_8u32f\_C4R, [1927](#)
- nppiCrossCorrFull\_Norm\_8u\_AC4RSfs, [1927](#)
- nppiCrossCorrFull\_Norm\_8u\_C1RSfs, [1928](#)
- nppiCrossCorrFull\_Norm\_8u\_C3RSfs, [1928](#)
- nppiCrossCorrFull\_Norm\_8u\_C4RSfs, [1929](#)
- crosscorrfullnormlevel
  - nppiCrossCorrFull\_NormLevel\_16u32f\_-AC4R, [1959](#)
  - nppiCrossCorrFull\_NormLevel\_16u32f\_C1R, [1959](#)
  - nppiCrossCorrFull\_NormLevel\_16u32f\_C3R, [1959](#)
  - nppiCrossCorrFull\_NormLevel\_16u32f\_C4R, [1960](#)
  - nppiCrossCorrFull\_NormLevel\_32f\_AC4R, [1960](#)
  - nppiCrossCorrFull\_NormLevel\_32f\_C1R, [1961](#)
  - nppiCrossCorrFull\_NormLevel\_32f\_C3R, [1961](#)
  - nppiCrossCorrFull\_NormLevel\_32f\_C4R, [1962](#)
  - nppiCrossCorrFull\_NormLevel\_8s32f\_AC4R, [1962](#)
  - nppiCrossCorrFull\_NormLevel\_8s32f\_C1R, [1963](#)
  - nppiCrossCorrFull\_NormLevel\_8s32f\_C3R, [1963](#)
  - nppiCrossCorrFull\_NormLevel\_8s32f\_C4R, [1964](#)
  - nppiCrossCorrFull\_NormLevel\_8u32f\_AC4R, [1964](#)
  - nppiCrossCorrFull\_NormLevel\_8u32f\_C1R, [1965](#)
  - nppiCrossCorrFull\_NormLevel\_8u32f\_C3R, [1965](#)
  - nppiCrossCorrFull\_NormLevel\_8u32f\_C4R, [1966](#)
  - nppiCrossCorrFull\_NormLevel\_8u\_AC4RSfs, [1966](#)
  - nppiCrossCorrFull\_NormLevel\_8u\_C1RSfs, [1967](#)
  - nppiCrossCorrFull\_NormLevel\_8u\_C3RSfs, [1967](#)
  - nppiCrossCorrFull\_NormLevel\_8u\_C4RSfs, [1968](#)
  - nppiFullNormLevelGetBufferHostSize\_-16u32f\_AC4R, [1968](#)
  - nppiFullNormLevelGetBufferHostSize\_-16u32f\_C1R, [1969](#)
  - nppiFullNormLevelGetBufferHostSize\_-16u32f\_C3R, [1969](#)
  - nppiFullNormLevelGetBufferHostSize\_-16u32f\_C4R, [1969](#)
  - nppiFullNormLevelGetBufferHostSize\_-AC4R, [1970](#)
  - nppiFullNormLevelGetBufferHostSize\_-C1R, [1970](#)
  - nppiFullNormLevelGetBufferHostSize\_-C3R, [1970](#)
  - nppiFullNormLevelGetBufferHostSize\_-C4R, [1970](#)
  - nppiFullNormLevelGetBufferHostSize\_-8s32f\_AC4R, [1971](#)
  - nppiFullNormLevelGetBufferHostSize\_-8s32f\_C1R, [1971](#)
  - nppiFullNormLevelGetBufferHostSize\_-8s32f\_C3R, [1971](#)
  - nppiFullNormLevelGetBufferHostSize\_-8s32f\_C4R, [1972](#)
  - nppiFullNormLevelGetBufferHostSize\_-8u32f\_AC4R, [1972](#)
  - nppiFullNormLevelGetBufferHostSize\_-8u32f\_C1R, [1972](#)
  - nppiFullNormLevelGetBufferHostSize\_-8u32f\_C3R, [1972](#)
  - nppiFullNormLevelGetBufferHostSize\_-8u32f\_C4R, [1973](#)
  - nppiFullNormLevelGetBufferHostSize\_8u\_-AC4RSfs, [1973](#)
  - nppiFullNormLevelGetBufferHostSize\_8u\_-C1RSfs, [1973](#)
  - nppiFullNormLevelGetBufferHostSize\_8u\_-C3RSfs, [1974](#)
  - nppiFullNormLevelGetBufferHostSize\_8u\_-C4RSfs, [1974](#)
- CrossCorrSame\_Norm, [1930](#)
- CrossCorrSame\_NormLevel, [1975](#)
- crosscorrnamenorm
  - nppiCrossCorrSame\_Norm\_16u32f\_AC4R, [1932](#)
  - nppiCrossCorrSame\_Norm\_16u32f\_C1R, [1932](#)
  - nppiCrossCorrSame\_Norm\_16u32f\_C3R, [1932](#)
  - nppiCrossCorrSame\_Norm\_16u32f\_C4R, [1933](#)
  - nppiCrossCorrSame\_Norm\_32f\_AC4R, [1933](#)
  - nppiCrossCorrSame\_Norm\_32f\_C1R, [1934](#)
  - nppiCrossCorrSame\_Norm\_32f\_C3R, [1934](#)
  - nppiCrossCorrSame\_Norm\_32f\_C4R, [1935](#)
  - nppiCrossCorrSame\_Norm\_8s32f\_AC4R, [1935](#)
  - nppiCrossCorrSame\_Norm\_8s32f\_C1R, [1935](#)
  - nppiCrossCorrSame\_Norm\_8s32f\_C3R, [1936](#)
  - nppiCrossCorrSame\_Norm\_8s32f\_C4R, [1936](#)
  - nppiCrossCorrSame\_Norm\_8u32f\_AC4R, [1937](#)



- nppiCrossCorrSame\_Norm\_8u32f\_C1R, [1937](#)
- nppiCrossCorrSame\_Norm\_8u32f\_C3R, [1938](#)
- nppiCrossCorrSame\_Norm\_8u32f\_C4R, [1938](#)
- nppiCrossCorrSame\_Norm\_8u\_AC4RSfs, [1938](#)
- nppiCrossCorrSame\_Norm\_8u\_C1RSfs, [1939](#)
- nppiCrossCorrSame\_Norm\_8u\_C3RSfs, [1939](#)
- nppiCrossCorrSame\_Norm\_8u\_C4RSfs, [1940](#)
- crosscorrnamenormlevel
  - nppiCrossCorrSame\_NormLevel\_16u32f\_-AC4R, [1979](#)
  - nppiCrossCorrSame\_NormLevel\_16u32f\_-C1R, [1979](#)
  - nppiCrossCorrSame\_NormLevel\_16u32f\_-C3R, [1979](#)
  - nppiCrossCorrSame\_NormLevel\_16u32f\_-C4R, [1980](#)
  - nppiCrossCorrSame\_NormLevel\_32f\_AC4R, [1980](#)
  - nppiCrossCorrSame\_NormLevel\_32f\_C1R, [1981](#)
  - nppiCrossCorrSame\_NormLevel\_32f\_C3R, [1981](#)
  - nppiCrossCorrSame\_NormLevel\_32f\_C4R, [1982](#)
  - nppiCrossCorrSame\_NormLevel\_8s32f\_-AC4R, [1982](#)
  - nppiCrossCorrSame\_NormLevel\_8s32f\_C1R, [1983](#)
  - nppiCrossCorrSame\_NormLevel\_8s32f\_C3R, [1983](#)
  - nppiCrossCorrSame\_NormLevel\_8s32f\_C4R, [1984](#)
  - nppiCrossCorrSame\_NormLevel\_8u32f\_-AC4R, [1984](#)
  - nppiCrossCorrSame\_NormLevel\_8u32f\_C1R, [1985](#)
  - nppiCrossCorrSame\_NormLevel\_8u32f\_C3R, [1985](#)
  - nppiCrossCorrSame\_NormLevel\_8u32f\_C4R, [1986](#)
  - nppiCrossCorrSame\_NormLevel\_8u\_-AC4RSfs, [1986](#)
  - nppiCrossCorrSame\_NormLevel\_8u\_C1RSfs, [1987](#)
  - nppiCrossCorrSame\_NormLevel\_8u\_C3RSfs, [1987](#)
  - nppiCrossCorrSame\_NormLevel\_8u\_C4RSfs, [1988](#)
  - nppiSameNormLevelGetBufferHostSize\_-16u32f\_AC4R, [1988](#)
  - nppiSameNormLevelGetBufferHostSize\_-16u32f\_C1R, [1989](#)
  - nppiSameNormLevelGetBufferHostSize\_-16u32f\_C3R, [1989](#)
  - nppiSameNormLevelGetBufferHostSize\_-16u32f\_C4R, [1989](#)
  - nppiSameNormLevelGetBufferHostSize\_-32f\_AC4R, [1990](#)
  - nppiSameNormLevelGetBufferHostSize\_-32f\_C1R, [1990](#)
  - nppiSameNormLevelGetBufferHostSize\_-32f\_C3R, [1990](#)
  - nppiSameNormLevelGetBufferHostSize\_-32f\_C4R, [1990](#)
  - nppiSameNormLevelGetBufferHostSize\_-8s32f\_AC4R, [1991](#)
  - nppiSameNormLevelGetBufferHostSize\_-8s32f\_C1R, [1991](#)
  - nppiSameNormLevelGetBufferHostSize\_-8s32f\_C3R, [1991](#)
  - nppiSameNormLevelGetBufferHostSize\_-8s32f\_C4R, [1992](#)
  - nppiSameNormLevelGetBufferHostSize\_-8u32f\_AC4R, [1992](#)
  - nppiSameNormLevelGetBufferHostSize\_-8u32f\_C1R, [1992](#)
  - nppiSameNormLevelGetBufferHostSize\_-8u32f\_C3R, [1992](#)
  - nppiSameNormLevelGetBufferHostSize\_-8u32f\_C4R, [1993](#)
  - nppiSameNormLevelGetBufferHostSize\_8u\_-AC4RSfs, [1993](#)
  - nppiSameNormLevelGetBufferHostSize\_8u\_-C1RSfs, [1993](#)
  - nppiSameNormLevelGetBufferHostSize\_8u\_-C3RSfs, [1994](#)
  - nppiSameNormLevelGetBufferHostSize\_8u\_-C4RSfs, [1994](#)
- CrossCorrValid, [1952](#)
- crosscorrvalid
  - nppiCrossCorrValid\_16u32f\_C1R, [1952](#)
  - nppiCrossCorrValid\_32f\_C1R, [1953](#)
  - nppiCrossCorrValid\_8s32f\_C1R, [1953](#)
  - nppiCrossCorrValid\_8u32f\_C1R, [1953](#)
- CrossCorrValid\_Norm, [1941](#)
- CrossCorrValid\_NormLevel, [1995](#)
- crosscorrvalidnorm
  - nppiCrossCorrValid\_Norm\_16u32f\_AC4R, [1943](#)
  - nppiCrossCorrValid\_Norm\_16u32f\_C1R, [1943](#)
  - nppiCrossCorrValid\_Norm\_16u32f\_C3R, [1943](#)
  - nppiCrossCorrValid\_Norm\_16u32f\_C4R, [1944](#)
  - nppiCrossCorrValid\_Norm\_32f\_AC4R, [1944](#)

- nppiCrossCorrValid\_Norm\_32f\_C1R, [1945](#)
- nppiCrossCorrValid\_Norm\_32f\_C3R, [1945](#)
- nppiCrossCorrValid\_Norm\_32f\_C4R, [1946](#)
- nppiCrossCorrValid\_Norm\_8s32f\_AC4R, [1946](#)
- nppiCrossCorrValid\_Norm\_8s32f\_C1R, [1946](#)
- nppiCrossCorrValid\_Norm\_8s32f\_C3R, [1947](#)
- nppiCrossCorrValid\_Norm\_8s32f\_C4R, [1947](#)
- nppiCrossCorrValid\_Norm\_8u32f\_AC4R, [1948](#)
- nppiCrossCorrValid\_Norm\_8u32f\_C1R, [1948](#)
- nppiCrossCorrValid\_Norm\_8u32f\_C3R, [1949](#)
- nppiCrossCorrValid\_Norm\_8u32f\_C4R, [1949](#)
- nppiCrossCorrValid\_Norm\_8u\_AC4RSfs, [1949](#)
- nppiCrossCorrValid\_Norm\_8u\_C1RSfs, [1950](#)
- nppiCrossCorrValid\_Norm\_8u\_C3RSfs, [1950](#)
- nppiCrossCorrValid\_Norm\_8u\_C4RSfs, [1951](#)
- crosscorrvalidnormlevel
  - nppiCrossCorrValid\_NormLevel\_16u32f\_AC4R, [1999](#)
  - nppiCrossCorrValid\_NormLevel\_16u32f\_C1R, [1999](#)
  - nppiCrossCorrValid\_NormLevel\_16u32f\_C3R, [1999](#)
  - nppiCrossCorrValid\_NormLevel\_16u32f\_C4R, [2000](#)
  - nppiCrossCorrValid\_NormLevel\_32f\_AC4R, [2000](#)
  - nppiCrossCorrValid\_NormLevel\_32f\_C1R, [2001](#)
  - nppiCrossCorrValid\_NormLevel\_32f\_C3R, [2001](#)
  - nppiCrossCorrValid\_NormLevel\_32f\_C4R, [2002](#)
  - nppiCrossCorrValid\_NormLevel\_8s32f\_AC4R, [2002](#)
  - nppiCrossCorrValid\_NormLevel\_8s32f\_C1R, [2003](#)
  - nppiCrossCorrValid\_NormLevel\_8s32f\_C3R, [2003](#)
  - nppiCrossCorrValid\_NormLevel\_8s32f\_C4R, [2004](#)
  - nppiCrossCorrValid\_NormLevel\_8u32f\_AC4R, [2004](#)
  - nppiCrossCorrValid\_NormLevel\_8u32f\_C1R, [2005](#)
  - nppiCrossCorrValid\_NormLevel\_8u32f\_C3R, [2005](#)
  - nppiCrossCorrValid\_NormLevel\_8u32f\_C4R, [2006](#)
  - nppiCrossCorrValid\_NormLevel\_8u\_-AC4RSfs, [2006](#)
- nppiCrossCorrValid\_NormLevel\_8u\_C1RSfs, [2007](#)
- nppiCrossCorrValid\_NormLevel\_8u\_C3RSfs, [2007](#)
- nppiCrossCorrValid\_NormLevel\_8u\_C4RSfs, [2008](#)
- nppiValidNormLevelGetBufferHostSize\_16u32f\_AC4R, [2008](#)
- nppiValidNormLevelGetBufferHostSize\_16u32f\_C1R, [2009](#)
- nppiValidNormLevelGetBufferHostSize\_16u32f\_C3R, [2009](#)
- nppiValidNormLevelGetBufferHostSize\_16u32f\_C4R, [2009](#)
- nppiValidNormLevelGetBufferHostSize\_32f\_AC4R, [2010](#)
- nppiValidNormLevelGetBufferHostSize\_32f\_C1R, [2010](#)
- nppiValidNormLevelGetBufferHostSize\_32f\_C3R, [2010](#)
- nppiValidNormLevelGetBufferHostSize\_32f\_C4R, [2010](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_AC4R, [2011](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_C1R, [2011](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_C3R, [2011](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_C4R, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_AC4R, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_C1R, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_C3R, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_C4R, [2013](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-AC4RSfs, [2013](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-C1RSfs, [2013](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-C3RSfs, [2014](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-C4RSfs, [2014](#)
- Cubrt, [2361](#)
- Data Exchange and Initialization, [719](#)
- Dilate3x3, [1381](#)
- Dilation, [1367](#)
- Div, [276](#), [2333](#)
- Div\_Round, [305](#), [2341](#)
- DivC, [140](#), [2285](#)

- DivCRev, [2292](#)
- Dot Product, [2548](#)
- DotProd, [1801](#)
- Duplicate Channel, [909](#)
- Erode, [1374](#)
- Erode3x3, [1387](#)
- Exp, [363](#), [2362](#)
- Filtering Functions, [941](#), [2441](#)
- Fixed Filters, [1151](#)
- fixed\_filters
  - nppiFilterPrewittHoriz\_16s\_AC4R, [1158](#)
  - nppiFilterPrewittHoriz\_16s\_C1R, [1158](#)
  - nppiFilterPrewittHoriz\_16s\_C3R, [1159](#)
  - nppiFilterPrewittHoriz\_16s\_C4R, [1159](#)
  - nppiFilterPrewittHoriz\_32f\_AC4R, [1159](#)
  - nppiFilterPrewittHoriz\_32f\_C1R, [1160](#)
  - nppiFilterPrewittHoriz\_32f\_C3R, [1160](#)
  - nppiFilterPrewittHoriz\_32f\_C4R, [1160](#)
  - nppiFilterPrewittHoriz\_8u\_AC4R, [1161](#)
  - nppiFilterPrewittHoriz\_8u\_C1R, [1161](#)
  - nppiFilterPrewittHoriz\_8u\_C3R, [1161](#)
  - nppiFilterPrewittHoriz\_8u\_C4R, [1162](#)
  - nppiFilterPrewittVert\_16s\_AC4R, [1162](#)
  - nppiFilterPrewittVert\_16s\_C1R, [1162](#)
  - nppiFilterPrewittVert\_16s\_C3R, [1163](#)
  - nppiFilterPrewittVert\_16s\_C4R, [1163](#)
  - nppiFilterPrewittVert\_32f\_AC4R, [1163](#)
  - nppiFilterPrewittVert\_32f\_C1R, [1164](#)
  - nppiFilterPrewittVert\_32f\_C3R, [1164](#)
  - nppiFilterPrewittVert\_32f\_C4R, [1164](#)
  - nppiFilterPrewittVert\_8u\_AC4R, [1165](#)
  - nppiFilterPrewittVert\_8u\_C1R, [1165](#)
  - nppiFilterPrewittVert\_8u\_C3R, [1165](#)
  - nppiFilterPrewittVert\_8u\_C4R, [1166](#)
  - nppiFilterScharrHoriz\_32f\_C1R, [1166](#)
  - nppiFilterScharrHoriz\_8s16s\_C1R, [1166](#)
  - nppiFilterScharrHoriz\_8u16s\_C1R, [1167](#)
  - nppiFilterScharrHorizBorder\_32f\_C1R, [1167](#)
  - nppiFilterScharrHorizBorder\_8s16s\_C1R, [1168](#)
  - nppiFilterScharrHorizBorder\_8u16s\_C1R, [1168](#)
  - nppiFilterScharrVert\_32f\_C1R, [1168](#)
  - nppiFilterScharrVert\_8s16s\_C1R, [1169](#)
  - nppiFilterScharrVert\_8u16s\_C1R, [1169](#)
  - nppiFilterScharrVertBorder\_32f\_C1R, [1169](#)
  - nppiFilterScharrVertBorder\_8s16s\_C1R, [1170](#)
  - nppiFilterScharrVertBorder\_8u16s\_C1R, [1170](#)
  - nppiFilterSobelHoriz\_16s\_AC4R, [1171](#)
  - nppiFilterSobelHoriz\_16s\_C1R, [1171](#)
  - nppiFilterSobelHoriz\_16s\_C3R, [1171](#)
  - nppiFilterSobelHoriz\_16s\_C4R, [1172](#)
  - nppiFilterSobelHoriz\_32f\_AC4R, [1172](#)
  - nppiFilterSobelHoriz\_32f\_C1R, [1172](#)
  - nppiFilterSobelHoriz\_32f\_C3R, [1173](#)
  - nppiFilterSobelHoriz\_32f\_C4R, [1173](#)
  - nppiFilterSobelHoriz\_8s16s\_C1R, [1173](#)
  - nppiFilterSobelHoriz\_8u16s\_C1R, [1174](#)
  - nppiFilterSobelHoriz\_8u\_AC4R, [1174](#)
  - nppiFilterSobelHoriz\_8u\_C1R, [1174](#)
  - nppiFilterSobelHoriz\_8u\_C3R, [1175](#)
  - nppiFilterSobelHoriz\_8u\_C4R, [1175](#)
  - nppiFilterSobelHorizMask\_32f\_C1R, [1175](#)
  - nppiFilterSobelHorizSecond\_32f\_C1R, [1176](#)
  - nppiFilterSobelHorizSecond\_8s16s\_C1R, [1176](#)
  - nppiFilterSobelHorizSecond\_8u16s\_C1R, [1177](#)
  - nppiFilterSobelVert\_16s\_AC4R, [1177](#)
  - nppiFilterSobelVert\_16s\_C1R, [1177](#)
  - nppiFilterSobelVert\_16s\_C3R, [1178](#)
  - nppiFilterSobelVert\_16s\_C4R, [1178](#)
  - nppiFilterSobelVert\_32f\_AC4R, [1178](#)
  - nppiFilterSobelVert\_32f\_C1R, [1179](#)
  - nppiFilterSobelVert\_32f\_C3R, [1179](#)
  - nppiFilterSobelVert\_32f\_C4R, [1179](#)
  - nppiFilterSobelVert\_8s16s\_C1R, [1180](#)
  - nppiFilterSobelVert\_8u16s\_C1R, [1180](#)
  - nppiFilterSobelVert\_8u\_AC4R, [1180](#)
  - nppiFilterSobelVert\_8u\_C1R, [1181](#)
  - nppiFilterSobelVert\_8u\_C3R, [1181](#)
  - nppiFilterSobelVert\_8u\_C4R, [1181](#)
  - nppiFilterSobelVertMask\_32f\_C1R, [1182](#)
- Fourier Transforms, [1364](#)
- Free, [2623](#)
- Geometry Transforms, [1183](#)
- GraphCut, [712](#)
- haarBuffer
  - NppiHaarBuffer, [2629](#)
- haarBufferSize
  - NppiHaarBuffer, [2629](#)
- height
  - NppiRect, [2632](#)
  - NppiSize, [2633](#)
- HistogramEven, [1854](#)
- HistogramRange, [1867](#)
- im
  - NPP\_ALIGN\_16, [2625](#)
  - NPP\_ALIGN\_8, [2627](#)
- Image Norms, [1597](#)
- Image Proximity, [1883](#)
- Image Quality Index, [2015](#)
- image\_1D\_linear\_filter

- [nppiFilterColumn32f\\_16s\\_AC4R](#), 999
- [nppiFilterColumn32f\\_16s\\_C1R](#), 1000
- [nppiFilterColumn32f\\_16s\\_C3R](#), 1000
- [nppiFilterColumn32f\\_16s\\_C4R](#), 1000
- [nppiFilterColumn32f\\_16u\\_AC4R](#), 1001
- [nppiFilterColumn32f\\_16u\\_C1R](#), 1001
- [nppiFilterColumn32f\\_16u\\_C3R](#), 1002
- [nppiFilterColumn32f\\_16u\\_C4R](#), 1002
- [nppiFilterColumn32f\\_8u\\_AC4R](#), 1003
- [nppiFilterColumn32f\\_8u\\_C1R](#), 1003
- [nppiFilterColumn32f\\_8u\\_C3R](#), 1004
- [nppiFilterColumn32f\\_8u\\_C4R](#), 1004
- [nppiFilterColumn\\_16s\\_AC4R](#), 1005
- [nppiFilterColumn\\_16s\\_C1R](#), 1005
- [nppiFilterColumn\\_16s\\_C3R](#), 1006
- [nppiFilterColumn\\_16s\\_C4R](#), 1006
- [nppiFilterColumn\\_16u\\_AC4R](#), 1007
- [nppiFilterColumn\\_16u\\_C1R](#), 1007
- [nppiFilterColumn\\_16u\\_C3R](#), 1008
- [nppiFilterColumn\\_16u\\_C4R](#), 1008
- [nppiFilterColumn\\_32f\\_AC4R](#), 1009
- [nppiFilterColumn\\_32f\\_C1R](#), 1009
- [nppiFilterColumn\\_32f\\_C3R](#), 1010
- [nppiFilterColumn\\_32f\\_C4R](#), 1010
- [nppiFilterColumn\\_64f\\_C1R](#), 1011
- [nppiFilterColumn\\_8u\\_AC4R](#), 1011
- [nppiFilterColumn\\_8u\\_C1R](#), 1012
- [nppiFilterColumn\\_8u\\_C3R](#), 1012
- [nppiFilterColumn\\_8u\\_C4R](#), 1013
- [nppiFilterRow32f\\_16s\\_AC4R](#), 1013
- [nppiFilterRow32f\\_16s\\_C1R](#), 1014
- [nppiFilterRow32f\\_16s\\_C3R](#), 1014
- [nppiFilterRow32f\\_16s\\_C4R](#), 1015
- [nppiFilterRow32f\\_16u\\_AC4R](#), 1015
- [nppiFilterRow32f\\_16u\\_C1R](#), 1016
- [nppiFilterRow32f\\_16u\\_C3R](#), 1016
- [nppiFilterRow32f\\_16u\\_C4R](#), 1017
- [nppiFilterRow32f\\_8u\\_AC4R](#), 1017
- [nppiFilterRow32f\\_8u\\_C1R](#), 1018
- [nppiFilterRow32f\\_8u\\_C3R](#), 1018
- [nppiFilterRow32f\\_8u\\_C4R](#), 1019
- [nppiFilterRow\\_16s\\_AC4R](#), 1019
- [nppiFilterRow\\_16s\\_C1R](#), 1020
- [nppiFilterRow\\_16s\\_C3R](#), 1020
- [nppiFilterRow\\_16s\\_C4R](#), 1021
- [nppiFilterRow\\_16u\\_AC4R](#), 1021
- [nppiFilterRow\\_16u\\_C1R](#), 1022
- [nppiFilterRow\\_16u\\_C3R](#), 1022
- [nppiFilterRow\\_16u\\_C4R](#), 1023
- [nppiFilterRow\\_32f\\_AC4R](#), 1023
- [nppiFilterRow\\_32f\\_C1R](#), 1024
- [nppiFilterRow\\_32f\\_C3R](#), 1024
- [nppiFilterRow\\_32f\\_C4R](#), 1025
- [nppiFilterRow\\_64f\\_C1R](#), 1025
- [nppiFilterRow\\_8u\\_AC4R](#), 1026
- [nppiFilterRow\\_8u\\_C1R](#), 1026
- [nppiFilterRow\\_8u\\_C3R](#), 1027
- [nppiFilterRow\\_8u\\_C4R](#), 1027
- [nppiFilterSobelCross\\_32f\\_C1R](#), 1028
- [nppiFilterSobelCross\\_8s16s\\_C1R](#), 1028
- [nppiFilterSobelCross\\_8u16s\\_C1R](#), 1029
- [nppiFilterSobelHorizBorder\\_16s\\_AC4R](#), 1029
- [nppiFilterSobelHorizBorder\\_16s\\_C1R](#), 1029
- [nppiFilterSobelHorizBorder\\_16s\\_C3R](#), 1030
- [nppiFilterSobelHorizBorder\\_16s\\_C4R](#), 1030
- [nppiFilterSobelHorizBorder\\_32f\\_AC4R](#), 1031
- [nppiFilterSobelHorizBorder\\_32f\\_C1R](#), 1031
- [nppiFilterSobelHorizBorder\\_32f\\_C3R](#), 1032
- [nppiFilterSobelHorizBorder\\_32f\\_C4R](#), 1032
- [nppiFilterSobelHorizBorder\\_8s16s\\_C1R](#), 1032
- [nppiFilterSobelHorizBorder\\_8u16s\\_C1R](#), 1033
- [nppiFilterSobelHorizBorder\\_8u\\_AC4R](#), 1033
- [nppiFilterSobelHorizBorder\\_8u\\_C1R](#), 1034
- [nppiFilterSobelHorizBorder\\_8u\\_C3R](#), 1034
- [nppiFilterSobelHorizBorder\\_8u\\_C4R](#), 1035
- [nppiFilterSobelHorizMaskBorder\\_32f\\_C1R](#), 1035
- [nppiFilterSobelHorizSecondBorder\\_32f\\_C1R](#), 1036
- [nppiFilterSobelHorizSecondBorder\\_8s16s\\_C1R](#), 1036
- [nppiFilterSobelHorizSecondBorder\\_8u16s\\_C1R](#), 1037
- [nppiFilterSobelVertBorder\\_16s\\_AC4R](#), 1037
- [nppiFilterSobelVertBorder\\_16s\\_C1R](#), 1037
- [nppiFilterSobelVertBorder\\_16s\\_C3R](#), 1038
- [nppiFilterSobelVertBorder\\_16s\\_C4R](#), 1038
- [nppiFilterSobelVertBorder\\_32f\\_AC4R](#), 1039
- [nppiFilterSobelVertBorder\\_32f\\_C1R](#), 1039
- [nppiFilterSobelVertBorder\\_32f\\_C3R](#), 1040
- [nppiFilterSobelVertBorder\\_32f\\_C4R](#), 1040
- [nppiFilterSobelVertBorder\\_8s16s\\_C1R](#), 1040
- [nppiFilterSobelVertBorder\\_8u16s\\_C1R](#), 1041
- [nppiFilterSobelVertBorder\\_8u\\_AC4R](#), 1041
- [nppiFilterSobelVertBorder\\_8u\\_C1R](#), 1042
- [nppiFilterSobelVertBorder\\_8u\\_C3R](#), 1042
- [nppiFilterSobelVertBorder\\_8u\\_C4R](#), 1043
- [nppiFilterSobelVertMaskBorder\\_32f\\_C1R](#), 1043
- [nppiFilterSobelVertSecond\\_32f\\_C1R](#), 1044
- [nppiFilterSobelVertSecond\\_8s16s\\_C1R](#), 1044
- [nppiFilterSobelVertSecond\\_8u16s\\_C1R](#), 1044
- [image\\_1D\\_window\\_sum](#)
- [nppiSumWindowColumn\\_16s32f\\_C1R](#), 1047
- [nppiSumWindowColumn\\_16s32f\\_C3R](#), 1048
- [nppiSumWindowColumn\\_16s32f\\_C4R](#), 1048

- nppiSumWindowColumn\_16u32f\_C1R, [1049](#)
- nppiSumWindowColumn\_16u32f\_C3R, [1049](#)
- nppiSumWindowColumn\_16u32f\_C4R, [1050](#)
- nppiSumWindowColumn\_8u32f\_C1R, [1050](#)
- nppiSumWindowColumn\_8u32f\_C3R, [1050](#)
- nppiSumWindowColumn\_8u32f\_C4R, [1051](#)
- nppiSumWindowRow\_16s32f\_C1R, [1051](#)
- nppiSumWindowRow\_16s32f\_C3R, [1052](#)
- nppiSumWindowRow\_16s32f\_C4R, [1052](#)
- nppiSumWindowRow\_16u32f\_C1R, [1053](#)
- nppiSumWindowRow\_16u32f\_C3R, [1053](#)
- nppiSumWindowRow\_16u32f\_C4R, [1054](#)
- nppiSumWindowRow\_8u32f\_C1R, [1054](#)
- nppiSumWindowRow\_8u32f\_C3R, [1055](#)
- nppiSumWindowRow\_8u32f\_C4R, [1055](#)
- image\_2D\_fixed\_linear\_filters
  - nppiFilterBox\_16s\_AC4R, [1113](#)
  - nppiFilterBox\_16s\_C1R, [1113](#)
  - nppiFilterBox\_16s\_C3R, [1114](#)
  - nppiFilterBox\_16s\_C4R, [1114](#)
  - nppiFilterBox\_16u\_AC4R, [1115](#)
  - nppiFilterBox\_16u\_C1R, [1115](#)
  - nppiFilterBox\_16u\_C3R, [1115](#)
  - nppiFilterBox\_16u\_C4R, [1116](#)
  - nppiFilterBox\_32f\_AC4R, [1116](#)
  - nppiFilterBox\_32f\_C1R, [1117](#)
  - nppiFilterBox\_32f\_C3R, [1117](#)
  - nppiFilterBox\_32f\_C4R, [1117](#)
  - nppiFilterBox\_64f\_C1R, [1118](#)
  - nppiFilterBox\_8u\_AC4R, [1118](#)
  - nppiFilterBox\_8u\_C1R, [1119](#)
  - nppiFilterBox\_8u\_C3R, [1119](#)
  - nppiFilterBox\_8u\_C4R, [1119](#)
- image\_abs
  - nppiAbs\_16s\_AC4IR, [321](#)
  - nppiAbs\_16s\_AC4R, [321](#)
  - nppiAbs\_16s\_C1IR, [321](#)
  - nppiAbs\_16s\_C1R, [322](#)
  - nppiAbs\_16s\_C3IR, [322](#)
  - nppiAbs\_16s\_C3R, [322](#)
  - nppiAbs\_16s\_C4IR, [323](#)
  - nppiAbs\_16s\_C4R, [323](#)
  - nppiAbs\_32f\_AC4IR, [323](#)
  - nppiAbs\_32f\_AC4R, [324](#)
  - nppiAbs\_32f\_C1IR, [324](#)
  - nppiAbs\_32f\_C1R, [324](#)
  - nppiAbs\_32f\_C3IR, [325](#)
  - nppiAbs\_32f\_C3R, [325](#)
  - nppiAbs\_32f\_C4IR, [325](#)
  - nppiAbs\_32f\_C4R, [326](#)
- image\_absdiff
  - nppiAbsDiff\_16u\_C1R, [327](#)
  - nppiAbsDiff\_32f\_C1R, [328](#)
  - nppiAbsDiff\_8u\_C1R, [328](#)
  - nppiAbsDiff\_8u\_C3R, [328](#)
  - nppiAbsDiff\_8u\_C4R, [329](#)
- image\_absdiffc
  - nppiAbsDiffC\_16u\_C1R, [166](#)
  - nppiAbsDiffC\_32f\_C1R, [166](#)
  - nppiAbsDiffC\_8u\_C1R, [167](#)
- image\_add
  - nppiAdd\_16s\_AC4IRSfs, [173](#)
  - nppiAdd\_16s\_AC4RSfs, [173](#)
  - nppiAdd\_16s\_C1IRSfs, [174](#)
  - nppiAdd\_16s\_C1RSfs, [174](#)
  - nppiAdd\_16s\_C3IRSfs, [175](#)
  - nppiAdd\_16s\_C3RSfs, [175](#)
  - nppiAdd\_16s\_C4IRSfs, [175](#)
  - nppiAdd\_16s\_C4RSfs, [176](#)
  - nppiAdd\_16sc\_AC4IRSfs, [176](#)
  - nppiAdd\_16sc\_AC4RSfs, [177](#)
  - nppiAdd\_16sc\_C1IRSfs, [177](#)
  - nppiAdd\_16sc\_C1RSfs, [177](#)
  - nppiAdd\_16sc\_C3IRSfs, [178](#)
  - nppiAdd\_16sc\_C3RSfs, [178](#)
  - nppiAdd\_16u\_AC4IRSfs, [179](#)
  - nppiAdd\_16u\_AC4RSfs, [179](#)
  - nppiAdd\_16u\_C1IRSfs, [180](#)
  - nppiAdd\_16u\_C1RSfs, [180](#)
  - nppiAdd\_16u\_C3IRSfs, [180](#)
  - nppiAdd\_16u\_C3RSfs, [181](#)
  - nppiAdd\_16u\_C4IRSfs, [181](#)
  - nppiAdd\_16u\_C4RSfs, [182](#)
  - nppiAdd\_32f\_AC4IR, [182](#)
  - nppiAdd\_32f\_AC4R, [182](#)
  - nppiAdd\_32f\_C1IR, [183](#)
  - nppiAdd\_32f\_C1R, [183](#)
  - nppiAdd\_32f\_C3IR, [184](#)
  - nppiAdd\_32f\_C3R, [184](#)
  - nppiAdd\_32f\_C4IR, [184](#)
  - nppiAdd\_32f\_C4R, [185](#)
  - nppiAdd\_32fc\_AC4IR, [185](#)
  - nppiAdd\_32fc\_AC4R, [185](#)
  - nppiAdd\_32fc\_C1IR, [186](#)
  - nppiAdd\_32fc\_C1R, [186](#)
  - nppiAdd\_32fc\_C3IR, [187](#)
  - nppiAdd\_32fc\_C3R, [187](#)
  - nppiAdd\_32fc\_C4IR, [187](#)
  - nppiAdd\_32fc\_C4R, [188](#)
  - nppiAdd\_32s\_C1IRSfs, [188](#)
  - nppiAdd\_32s\_C1R, [189](#)
  - nppiAdd\_32s\_C1RSfs, [189](#)
  - nppiAdd\_32s\_C3IRSfs, [189](#)
  - nppiAdd\_32s\_C3RSfs, [190](#)
  - nppiAdd\_32sc\_AC4IRSfs, [190](#)
  - nppiAdd\_32sc\_AC4RSfs, [191](#)
  - nppiAdd\_32sc\_C1IRSfs, [191](#)
  - nppiAdd\_32sc\_C1RSfs, [191](#)



- nppiAdd\_32sc\_C3IRSfs, 192
- nppiAdd\_32sc\_C3RSfs, 192
- nppiAdd\_8u\_AC4IRSfs, 193
- nppiAdd\_8u\_AC4RSfs, 193
- nppiAdd\_8u\_C1IRSfs, 194
- nppiAdd\_8u\_C1RSfs, 194
- nppiAdd\_8u\_C3IRSfs, 194
- nppiAdd\_8u\_C3RSfs, 195
- nppiAdd\_8u\_C4IRSfs, 195
- nppiAdd\_8u\_C4RSfs, 196
- image\_addc
  - nppiAddC\_16s\_AC4IRSfs, 60
  - nppiAddC\_16s\_AC4RSfs, 60
  - nppiAddC\_16s\_C1IRSfs, 60
  - nppiAddC\_16s\_C1RSfs, 61
  - nppiAddC\_16s\_C3IRSfs, 61
  - nppiAddC\_16s\_C3RSfs, 61
  - nppiAddC\_16s\_C4IRSfs, 62
  - nppiAddC\_16s\_C4RSfs, 62
  - nppiAddC\_16sc\_AC4IRSfs, 63
  - nppiAddC\_16sc\_AC4RSfs, 63
  - nppiAddC\_16sc\_C1IRSfs, 63
  - nppiAddC\_16sc\_C1RSfs, 64
  - nppiAddC\_16sc\_C3IRSfs, 64
  - nppiAddC\_16sc\_C3RSfs, 65
  - nppiAddC\_16u\_AC4IRSfs, 65
  - nppiAddC\_16u\_AC4RSfs, 65
  - nppiAddC\_16u\_C1IRSfs, 66
  - nppiAddC\_16u\_C1RSfs, 66
  - nppiAddC\_16u\_C3IRSfs, 67
  - nppiAddC\_16u\_C3RSfs, 67
  - nppiAddC\_16u\_C4IRSfs, 67
  - nppiAddC\_16u\_C4RSfs, 68
  - nppiAddC\_32f\_AC4IR, 68
  - nppiAddC\_32f\_AC4R, 68
  - nppiAddC\_32f\_C1IR, 69
  - nppiAddC\_32f\_C1R, 69
  - nppiAddC\_32f\_C3IR, 69
  - nppiAddC\_32f\_C3R, 70
  - nppiAddC\_32f\_C4IR, 70
  - nppiAddC\_32f\_C4R, 70
  - nppiAddC\_32fc\_AC4IR, 71
  - nppiAddC\_32fc\_AC4R, 71
  - nppiAddC\_32fc\_C1IR, 71
  - nppiAddC\_32fc\_C1R, 72
  - nppiAddC\_32fc\_C3IR, 72
  - nppiAddC\_32fc\_C3R, 72
  - nppiAddC\_32fc\_C4IR, 73
  - nppiAddC\_32fc\_C4R, 73
  - nppiAddC\_32s\_C1IRSfs, 74
  - nppiAddC\_32s\_C1RSfs, 74
  - nppiAddC\_32s\_C3IRSfs, 74
  - nppiAddC\_32s\_C3RSfs, 75
  - nppiAddC\_32sc\_AC4IRSfs, 75
  - nppiAddC\_32sc\_AC4RSfs, 75
  - nppiAddC\_32sc\_C1IRSfs, 76
  - nppiAddC\_32sc\_C1RSfs, 76
  - nppiAddC\_32sc\_C3IRSfs, 77
  - nppiAddC\_32sc\_C3RSfs, 77
  - nppiAddC\_8u\_AC4IRSfs, 77
  - nppiAddC\_8u\_AC4RSfs, 78
  - nppiAddC\_8u\_C1IRSfs, 78
  - nppiAddC\_8u\_C1RSfs, 79
  - nppiAddC\_8u\_C3IRSfs, 79
  - nppiAddC\_8u\_C3RSfs, 79
  - nppiAddC\_8u\_C4IRSfs, 80
  - nppiAddC\_8u\_C4RSfs, 80
- image\_addproduct
  - nppiAddProduct\_16u32f\_C1IMR, 200
  - nppiAddProduct\_16u32f\_C1IR, 201
  - nppiAddProduct\_32f\_C1IMR, 201
  - nppiAddProduct\_32f\_C1IR, 202
  - nppiAddProduct\_8u32f\_C1IMR, 202
  - nppiAddProduct\_8u32f\_C1IR, 202
- image\_addsquare
  - nppiAddSquare\_16u32f\_C1IMR, 197
  - nppiAddSquare\_16u32f\_C1IR, 198
  - nppiAddSquare\_32f\_C1IMR, 198
  - nppiAddSquare\_32f\_C1IR, 198
  - nppiAddSquare\_8u32f\_C1IMR, 199
  - nppiAddSquare\_8u32f\_C1IR, 199
- image\_addweighted
  - nppiAddWeighted\_16u32f\_C1IMR, 204
  - nppiAddWeighted\_16u32f\_C1IR, 205
  - nppiAddWeighted\_32f\_C1IMR, 205
  - nppiAddWeighted\_32f\_C1IR, 206
  - nppiAddWeighted\_8u32f\_C1IMR, 206
  - nppiAddWeighted\_8u32f\_C1IR, 206
- image\_affine\_transform
  - nppiGetAffineBound, 1276
  - nppiGetAffineQuad, 1276
  - nppiGetAffineTransform, 1277
  - nppiWarpAffine\_16u\_AC4R, 1277
  - nppiWarpAffine\_16u\_C1R, 1278
  - nppiWarpAffine\_16u\_C3R, 1278
  - nppiWarpAffine\_16u\_C4R, 1279
  - nppiWarpAffine\_16u\_P3R, 1279
  - nppiWarpAffine\_16u\_P4R, 1280
  - nppiWarpAffine\_32f\_AC4R, 1280
  - nppiWarpAffine\_32f\_C1R, 1281
  - nppiWarpAffine\_32f\_C3R, 1281
  - nppiWarpAffine\_32f\_C4R, 1282
  - nppiWarpAffine\_32f\_P3R, 1282
  - nppiWarpAffine\_32f\_P4R, 1283
  - nppiWarpAffine\_32s\_AC4R, 1283
  - nppiWarpAffine\_32s\_C1R, 1284
  - nppiWarpAffine\_32s\_C3R, 1284
  - nppiWarpAffine\_32s\_C4R, 1285

- [nppiWarpAffine\\_32s\\_P3R](#), [1285](#)
- [nppiWarpAffine\\_32s\\_P4R](#), [1286](#)
- [nppiWarpAffine\\_64f\\_AC4R](#), [1286](#)
- [nppiWarpAffine\\_64f\\_C1R](#), [1287](#)
- [nppiWarpAffine\\_64f\\_C3R](#), [1287](#)
- [nppiWarpAffine\\_64f\\_C4R](#), [1288](#)
- [nppiWarpAffine\\_64f\\_P3R](#), [1288](#)
- [nppiWarpAffine\\_64f\\_P4R](#), [1289](#)
- [nppiWarpAffine\\_8u\\_AC4R](#), [1289](#)
- [nppiWarpAffine\\_8u\\_C1R](#), [1290](#)
- [nppiWarpAffine\\_8u\\_C3R](#), [1290](#)
- [nppiWarpAffine\\_8u\\_C4R](#), [1291](#)
- [nppiWarpAffine\\_8u\\_P3R](#), [1291](#)
- [nppiWarpAffine\\_8u\\_P4R](#), [1292](#)
- [nppiWarpAffineBack\\_16u\\_AC4R](#), [1292](#)
- [nppiWarpAffineBack\\_16u\\_C1R](#), [1293](#)
- [nppiWarpAffineBack\\_16u\\_C3R](#), [1293](#)
- [nppiWarpAffineBack\\_16u\\_C4R](#), [1294](#)
- [nppiWarpAffineBack\\_16u\\_P3R](#), [1294](#)
- [nppiWarpAffineBack\\_16u\\_P4R](#), [1295](#)
- [nppiWarpAffineBack\\_32f\\_AC4R](#), [1295](#)
- [nppiWarpAffineBack\\_32f\\_C1R](#), [1296](#)
- [nppiWarpAffineBack\\_32f\\_C3R](#), [1296](#)
- [nppiWarpAffineBack\\_32f\\_C4R](#), [1297](#)
- [nppiWarpAffineBack\\_32f\\_P3R](#), [1297](#)
- [nppiWarpAffineBack\\_32f\\_P4R](#), [1298](#)
- [nppiWarpAffineBack\\_32s\\_AC4R](#), [1298](#)
- [nppiWarpAffineBack\\_32s\\_C1R](#), [1299](#)
- [nppiWarpAffineBack\\_32s\\_C3R](#), [1299](#)
- [nppiWarpAffineBack\\_32s\\_C4R](#), [1300](#)
- [nppiWarpAffineBack\\_32s\\_P3R](#), [1300](#)
- [nppiWarpAffineBack\\_32s\\_P4R](#), [1301](#)
- [nppiWarpAffineBack\\_8u\\_AC4R](#), [1301](#)
- [nppiWarpAffineBack\\_8u\\_C1R](#), [1302](#)
- [nppiWarpAffineBack\\_8u\\_C3R](#), [1302](#)
- [nppiWarpAffineBack\\_8u\\_C4R](#), [1303](#)
- [nppiWarpAffineBack\\_8u\\_P3R](#), [1303](#)
- [nppiWarpAffineBack\\_8u\\_P4R](#), [1304](#)
- [nppiWarpAffineQuad\\_16u\\_AC4R](#), [1304](#)
- [nppiWarpAffineQuad\\_16u\\_C1R](#), [1305](#)
- [nppiWarpAffineQuad\\_16u\\_C3R](#), [1305](#)
- [nppiWarpAffineQuad\\_16u\\_C4R](#), [1306](#)
- [nppiWarpAffineQuad\\_16u\\_P3R](#), [1306](#)
- [nppiWarpAffineQuad\\_16u\\_P4R](#), [1307](#)
- [nppiWarpAffineQuad\\_32f\\_AC4R](#), [1307](#)
- [nppiWarpAffineQuad\\_32f\\_C1R](#), [1308](#)
- [nppiWarpAffineQuad\\_32f\\_C3R](#), [1308](#)
- [nppiWarpAffineQuad\\_32f\\_C4R](#), [1309](#)
- [nppiWarpAffineQuad\\_32f\\_P3R](#), [1309](#)
- [nppiWarpAffineQuad\\_32f\\_P4R](#), [1310](#)
- [nppiWarpAffineQuad\\_32s\\_AC4R](#), [1310](#)
- [nppiWarpAffineQuad\\_32s\\_C1R](#), [1311](#)
- [nppiWarpAffineQuad\\_32s\\_C3R](#), [1311](#)
- [nppiWarpAffineQuad\\_32s\\_C4R](#), [1312](#)
- [nppiWarpAffineQuad\\_32s\\_P3R](#), [1312](#)
- [nppiWarpAffineQuad\\_32s\\_P4R](#), [1313](#)
- [nppiWarpAffineQuad\\_8u\\_AC4R](#), [1313](#)
- [nppiWarpAffineQuad\\_8u\\_C1R](#), [1314](#)
- [nppiWarpAffineQuad\\_8u\\_C3R](#), [1314](#)
- [nppiWarpAffineQuad\\_8u\\_C4R](#), [1315](#)
- [nppiWarpAffineQuad\\_8u\\_P3R](#), [1315](#)
- [nppiWarpAffineQuad\\_8u\\_P4R](#), [1316](#)
- [image\\_alphacomp](#)
  - [nppiAlphaComp\\_16s\\_AC1R](#), [489](#)
  - [nppiAlphaComp\\_16u\\_AC1R](#), [489](#)
  - [nppiAlphaComp\\_16u\\_AC4R](#), [490](#)
  - [nppiAlphaComp\\_32f\\_AC1R](#), [490](#)
  - [nppiAlphaComp\\_32f\\_AC4R](#), [491](#)
  - [nppiAlphaComp\\_32s\\_AC1R](#), [491](#)
  - [nppiAlphaComp\\_32s\\_AC4R](#), [491](#)
  - [nppiAlphaComp\\_32u\\_AC1R](#), [492](#)
  - [nppiAlphaComp\\_32u\\_AC4R](#), [492](#)
  - [nppiAlphaComp\\_8s\\_AC1R](#), [493](#)
  - [nppiAlphaComp\\_8u\\_AC1R](#), [493](#)
  - [nppiAlphaComp\\_8u\\_AC4R](#), [494](#)
- [image\\_alphacompc](#)
  - [nppiAlphaCompC\\_16s\\_C1R](#), [474](#)
  - [nppiAlphaCompC\\_16u\\_AC4R](#), [474](#)
  - [nppiAlphaCompC\\_16u\\_C1R](#), [475](#)
  - [nppiAlphaCompC\\_16u\\_C3R](#), [475](#)
  - [nppiAlphaCompC\\_16u\\_C4R](#), [476](#)
  - [nppiAlphaCompC\\_32f\\_C1R](#), [476](#)
  - [nppiAlphaCompC\\_32s\\_C1R](#), [477](#)
  - [nppiAlphaCompC\\_32u\\_C1R](#), [477](#)
  - [nppiAlphaCompC\\_8s\\_C1R](#), [478](#)
  - [nppiAlphaCompC\\_8u\\_AC4R](#), [478](#)
  - [nppiAlphaCompC\\_8u\\_C1R](#), [479](#)
  - [nppiAlphaCompC\\_8u\\_C3R](#), [479](#)
  - [nppiAlphaCompC\\_8u\\_C4R](#), [480](#)
- [image\\_alphapremul](#)
  - [nppiAlphaPremul\\_16u\\_AC4IR](#), [495](#)
  - [nppiAlphaPremul\\_16u\\_AC4R](#), [495](#)
  - [nppiAlphaPremul\\_8u\\_AC4IR](#), [496](#)
  - [nppiAlphaPremul\\_8u\\_AC4R](#), [496](#)
- [image\\_alphapremulc](#)
  - [nppiAlphaPremulC\\_16u\\_AC4IR](#), [482](#)
  - [nppiAlphaPremulC\\_16u\\_AC4R](#), [482](#)
  - [nppiAlphaPremulC\\_16u\\_C1IR](#), [483](#)
  - [nppiAlphaPremulC\\_16u\\_C1R](#), [483](#)
  - [nppiAlphaPremulC\\_16u\\_C3IR](#), [483](#)
  - [nppiAlphaPremulC\\_16u\\_C3R](#), [484](#)
  - [nppiAlphaPremulC\\_16u\\_C4IR](#), [484](#)
  - [nppiAlphaPremulC\\_16u\\_C4R](#), [484](#)
  - [nppiAlphaPremulC\\_8u\\_AC4IR](#), [485](#)
  - [nppiAlphaPremulC\\_8u\\_AC4R](#), [485](#)
  - [nppiAlphaPremulC\\_8u\\_C1IR](#), [485](#)
  - [nppiAlphaPremulC\\_8u\\_C1R](#), [486](#)
  - [nppiAlphaPremulC\\_8u\\_C3IR](#), [486](#)

- nppiAlphaPremulC\_8u\_C3R, [486](#)
  - nppiAlphaPremulC\_8u\_C4IR, [487](#)
  - nppiAlphaPremulC\_8u\_C4R, [487](#)
- image\_and
  - nppiAnd\_16u\_AC4IR, [434](#)
  - nppiAnd\_16u\_AC4R, [434](#)
  - nppiAnd\_16u\_C1IR, [434](#)
  - nppiAnd\_16u\_C1R, [435](#)
  - nppiAnd\_16u\_C3IR, [435](#)
  - nppiAnd\_16u\_C3R, [435](#)
  - nppiAnd\_16u\_C4IR, [436](#)
  - nppiAnd\_16u\_C4R, [436](#)
  - nppiAnd\_32s\_AC4IR, [437](#)
  - nppiAnd\_32s\_AC4R, [437](#)
  - nppiAnd\_32s\_C1IR, [437](#)
  - nppiAnd\_32s\_C1R, [438](#)
  - nppiAnd\_32s\_C3IR, [438](#)
  - nppiAnd\_32s\_C3R, [438](#)
  - nppiAnd\_32s\_C4IR, [439](#)
  - nppiAnd\_32s\_C4R, [439](#)
  - nppiAnd\_8u\_AC4IR, [440](#)
  - nppiAnd\_8u\_AC4R, [440](#)
  - nppiAnd\_8u\_C1IR, [440](#)
  - nppiAnd\_8u\_C1R, [441](#)
  - nppiAnd\_8u\_C3IR, [441](#)
  - nppiAnd\_8u\_C3R, [441](#)
  - nppiAnd\_8u\_C4IR, [442](#)
  - nppiAnd\_8u\_C4R, [442](#)
- image\_andc
  - nppiAndC\_16u\_AC4IR, [373](#)
  - nppiAndC\_16u\_AC4R, [373](#)
  - nppiAndC\_16u\_C1IR, [373](#)
  - nppiAndC\_16u\_C1R, [374](#)
  - nppiAndC\_16u\_C3IR, [374](#)
  - nppiAndC\_16u\_C3R, [374](#)
  - nppiAndC\_16u\_C4IR, [375](#)
  - nppiAndC\_16u\_C4R, [375](#)
  - nppiAndC\_32s\_AC4IR, [375](#)
  - nppiAndC\_32s\_AC4R, [376](#)
  - nppiAndC\_32s\_C1IR, [376](#)
  - nppiAndC\_32s\_C1R, [376](#)
  - nppiAndC\_32s\_C3IR, [377](#)
  - nppiAndC\_32s\_C3R, [377](#)
  - nppiAndC\_32s\_C4IR, [377](#)
  - nppiAndC\_32s\_C4R, [378](#)
  - nppiAndC\_8u\_AC4IR, [378](#)
  - nppiAndC\_8u\_AC4R, [378](#)
  - nppiAndC\_8u\_C1IR, [379](#)
  - nppiAndC\_8u\_C1R, [379](#)
  - nppiAndC\_8u\_C3IR, [379](#)
  - nppiAndC\_8u\_C3R, [380](#)
  - nppiAndC\_8u\_C4IR, [380](#)
  - nppiAndC\_8u\_C4R, [380](#)
- image\_average\_error
  - nppiAverageError\_16s\_C1R, [2050](#)
  - nppiAverageError\_16s\_C2R, [2051](#)
  - nppiAverageError\_16s\_C3R, [2051](#)
  - nppiAverageError\_16s\_C4R, [2052](#)
  - nppiAverageError\_16sc\_C1R, [2052](#)
  - nppiAverageError\_16sc\_C2R, [2052](#)
  - nppiAverageError\_16sc\_C3R, [2053](#)
  - nppiAverageError\_16sc\_C4R, [2053](#)
  - nppiAverageError\_16u\_C1R, [2054](#)
  - nppiAverageError\_16u\_C2R, [2054](#)
  - nppiAverageError\_16u\_C3R, [2055](#)
  - nppiAverageError\_16u\_C4R, [2055](#)
  - nppiAverageError\_32f\_C1R, [2055](#)
  - nppiAverageError\_32f\_C2R, [2056](#)
  - nppiAverageError\_32f\_C3R, [2056](#)
  - nppiAverageError\_32f\_C4R, [2057](#)
  - nppiAverageError\_32fc\_C1R, [2057](#)
  - nppiAverageError\_32fc\_C2R, [2058](#)
  - nppiAverageError\_32fc\_C3R, [2058](#)
  - nppiAverageError\_32fc\_C4R, [2059](#)
  - nppiAverageError\_32s\_C1R, [2059](#)
  - nppiAverageError\_32s\_C2R, [2059](#)
  - nppiAverageError\_32s\_C3R, [2060](#)
  - nppiAverageError\_32s\_C4R, [2060](#)
  - nppiAverageError\_32sc\_C1R, [2061](#)
  - nppiAverageError\_32sc\_C2R, [2061](#)
  - nppiAverageError\_32sc\_C3R, [2062](#)
  - nppiAverageError\_32sc\_C4R, [2062](#)
  - nppiAverageError\_32u\_C1R, [2062](#)
  - nppiAverageError\_32u\_C2R, [2063](#)
  - nppiAverageError\_32u\_C3R, [2063](#)
  - nppiAverageError\_32u\_C4R, [2064](#)
  - nppiAverageError\_64f\_C1R, [2064](#)
  - nppiAverageError\_64f\_C2R, [2065](#)
  - nppiAverageError\_64f\_C3R, [2065](#)
  - nppiAverageError\_64f\_C4R, [2066](#)
  - nppiAverageError\_8s\_C1R, [2066](#)
  - nppiAverageError\_8s\_C2R, [2066](#)
  - nppiAverageError\_8s\_C3R, [2067](#)
  - nppiAverageError\_8s\_C4R, [2067](#)
  - nppiAverageError\_8u\_C1R, [2068](#)
  - nppiAverageError\_8u\_C2R, [2068](#)
  - nppiAverageError\_8u\_C3R, [2069](#)
  - nppiAverageError\_8u\_C4R, [2069](#)
- image\_average\_relative\_error
  - nppiAverageRelativeError\_16s\_C1R, [2097](#)
  - nppiAverageRelativeError\_16s\_C2R, [2098](#)
  - nppiAverageRelativeError\_16s\_C3R, [2098](#)
  - nppiAverageRelativeError\_16s\_C4R, [2099](#)
  - nppiAverageRelativeError\_16sc\_C1R, [2099](#)
  - nppiAverageRelativeError\_16sc\_C2R, [2100](#)
  - nppiAverageRelativeError\_16sc\_C3R, [2100](#)
  - nppiAverageRelativeError\_16sc\_C4R, [2100](#)
  - nppiAverageRelativeError\_16u\_C1R, [2101](#)



- nppiAverageRelativeError\_16u\_C2R, [2101](#)
- nppiAverageRelativeError\_16u\_C3R, [2102](#)
- nppiAverageRelativeError\_16u\_C4R, [2102](#)
- nppiAverageRelativeError\_32f\_C1R, [2103](#)
- nppiAverageRelativeError\_32f\_C2R, [2103](#)
- nppiAverageRelativeError\_32f\_C3R, [2104](#)
- nppiAverageRelativeError\_32f\_C4R, [2104](#)
- nppiAverageRelativeError\_32fc\_C1R, [2105](#)
- nppiAverageRelativeError\_32fc\_C2R, [2105](#)
- nppiAverageRelativeError\_32fc\_C3R, [2105](#)
- nppiAverageRelativeError\_32fc\_C4R, [2106](#)
- nppiAverageRelativeError\_32s\_C1R, [2106](#)
- nppiAverageRelativeError\_32s\_C2R, [2107](#)
- nppiAverageRelativeError\_32s\_C3R, [2107](#)
- nppiAverageRelativeError\_32s\_C4R, [2108](#)
- nppiAverageRelativeError\_32sc\_C1R, [2108](#)
- nppiAverageRelativeError\_32sc\_C2R, [2109](#)
- nppiAverageRelativeError\_32sc\_C3R, [2109](#)
- nppiAverageRelativeError\_32sc\_C4R, [2110](#)
- nppiAverageRelativeError\_32u\_C1R, [2110](#)
- nppiAverageRelativeError\_32u\_C2R, [2110](#)
- nppiAverageRelativeError\_32u\_C3R, [2111](#)
- nppiAverageRelativeError\_32u\_C4R, [2111](#)
- nppiAverageRelativeError\_64f\_C1R, [2112](#)
- nppiAverageRelativeError\_64f\_C2R, [2112](#)
- nppiAverageRelativeError\_64f\_C3R, [2113](#)
- nppiAverageRelativeError\_64f\_C4R, [2113](#)
- nppiAverageRelativeError\_8s\_C1R, [2114](#)
- nppiAverageRelativeError\_8s\_C2R, [2114](#)
- nppiAverageRelativeError\_8s\_C3R, [2115](#)
- nppiAverageRelativeError\_8s\_C4R, [2115](#)
- nppiAverageRelativeError\_8u\_C1R, [2115](#)
- nppiAverageRelativeError\_8u\_C2R, [2116](#)
- nppiAverageRelativeError\_8u\_C3R, [2116](#)
- nppiAverageRelativeError\_8u\_C4R, [2117](#)
- image\_color\_gamma\_correction
  - nppiGammaFwd\_8u\_AC4IR, [607](#)
  - nppiGammaFwd\_8u\_AC4R, [607](#)
  - nppiGammaFwd\_8u\_C3IR, [607](#)
  - nppiGammaFwd\_8u\_C3R, [608](#)
  - nppiGammaFwd\_8u\_IP3R, [608](#)
  - nppiGammaFwd\_8u\_P3R, [608](#)
  - nppiGammaInv\_8u\_AC4IR, [609](#)
  - nppiGammaInv\_8u\_AC4R, [609](#)
  - nppiGammaInv\_8u\_C3IR, [609](#)
  - nppiGammaInv\_8u\_C3R, [610](#)
  - nppiGammaInv\_8u\_IP3R, [610](#)
  - nppiGammaInv\_8u\_P3R, [610](#)
- image\_color\_model\_conversion
  - nppiBGRToCbYCr422\_709HDTV\_8u\_-AC4C2R, [525](#)
  - nppiBGRToCbYCr422\_709HDTV\_8u\_-C3C2R, [525](#)
  - nppiBGRToCbYCr422\_8u\_AC4C2R, [525](#)
  - nppiBGRToHLS\_8u\_AC4P4R, [526](#)
  - nppiBGRToHLS\_8u\_AC4R, [526](#)
  - nppiBGRToHLS\_8u\_AP4C4R, [526](#)
  - nppiBGRToHLS\_8u\_AP4R, [527](#)
  - nppiBGRToHLS\_8u\_C3P3R, [527](#)
  - nppiBGRToHLS\_8u\_P3C3R, [527](#)
  - nppiBGRToHLS\_8u\_P3R, [528](#)
  - nppiBGRToLab\_8u\_C3R, [528](#)
  - nppiBGRToYCbCr411\_8u\_AC4P3R, [528](#)
  - nppiBGRToYCbCr411\_8u\_C3P3R, [529](#)
  - nppiBGRToYCbCr420\_709CSC\_8u\_-AC4P3R, [529](#)
  - nppiBGRToYCbCr420\_709CSC\_8u\_C3P3R, [530](#)
  - nppiBGRToYCbCr420\_709HDTV\_8u\_-AC4P3R, [530](#)
  - nppiBGRToYCbCr420\_8u\_AC4P3R, [530](#)
  - nppiBGRToYCbCr420\_8u\_C3P3R, [531](#)
  - nppiBGRToYCbCr422\_8u\_AC4C2R, [531](#)
  - nppiBGRToYCbCr422\_8u\_AC4P3R, [532](#)
  - nppiBGRToYCbCr422\_8u\_C3C2R, [532](#)
  - nppiBGRToYCbCr422\_8u\_C3P3R, [532](#)
  - nppiBGRToYCbCr\_8u\_AC4P3R, [533](#)
  - nppiBGRToYCbCr\_8u\_AC4P4R, [533](#)
  - nppiBGRToYCbCr\_8u\_C3P3R, [534](#)
  - nppiBGRToYCrCb420\_709CSC\_8u\_-AC4P3R, [534](#)
  - nppiBGRToYCrCb420\_709CSC\_8u\_C3P3R, [534](#)
  - nppiBGRToYCrCb420\_8u\_AC4P3R, [535](#)
  - nppiBGRToYCrCb420\_8u\_C3P3R, [535](#)
  - nppiBGRToYUV420\_8u\_AC4P3R, [536](#)
  - nppiBGRToYUV\_8u\_AC4P4R, [536](#)
  - nppiBGRToYUV\_8u\_AC4R, [536](#)
  - nppiBGRToYUV\_8u\_C3P3R, [537](#)
  - nppiBGRToYUV\_8u\_C3R, [537](#)
  - nppiBGRToYUV\_8u\_P3R, [537](#)
  - nppiCbYCr422ToBGR\_709HDTV\_8u\_-C2C3R, [538](#)
  - nppiCbYCr422ToBGR\_709HDTV\_8u\_-C2C4R, [538](#)
  - nppiCbYCr422ToBGR\_8u\_C2C4R, [539](#)
  - nppiCbYCr422ToRGB\_8u\_C2C3R, [539](#)
  - nppiColorToGray\_16s\_AC4C1R, [539](#)
  - nppiColorToGray\_16s\_C3C1R, [540](#)
  - nppiColorToGray\_16u\_AC4C1R, [540](#)
  - nppiColorToGray\_16u\_C3C1R, [540](#)
  - nppiColorToGray\_32f\_AC4C1R, [541](#)
  - nppiColorToGray\_32f\_C3C1R, [541](#)
  - nppiColorToGray\_8u\_AC4C1R, [542](#)
  - nppiColorToGray\_8u\_C3C1R, [542](#)
  - nppiHLSToBGR\_8u\_AC4P4R, [542](#)
  - nppiHLSToBGR\_8u\_AC4R, [543](#)
  - nppiHLSToBGR\_8u\_AP4C4R, [543](#)

- nppiHLSToBGR\_8u\_AP4R, 543
- nppiHLSToBGR\_8u\_C3P3R, 544
- nppiHLSToBGR\_8u\_P3C3R, 544
- nppiHLSToBGR\_8u\_P3R, 544
- nppiHLSToRGB\_8u\_AC4R, 545
- nppiHLSToRGB\_8u\_C3R, 545
- nppiHSVToRGB\_8u\_AC4R, 545
- nppiHSVToRGB\_8u\_C3R, 546
- nppiLabToBGR\_8u\_C3R, 546
- nppiLUVToRGB\_8u\_AC4R, 546
- nppiLUVToRGB\_8u\_C3R, 547
- nppiRGBToCbYCr422\_8u\_C3C2R, 547
- nppiRGBToCbYCr422Gamma\_8u\_C3C2R, 547
- nppiRGBToGray\_16s\_AC4C1R, 548
- nppiRGBToGray\_16s\_C3C1R, 548
- nppiRGBToGray\_16u\_AC4C1R, 548
- nppiRGBToGray\_16u\_C3C1R, 549
- nppiRGBToGray\_32f\_AC4C1R, 549
- nppiRGBToGray\_32f\_C3C1R, 549
- nppiRGBToGray\_8u\_AC4C1R, 550
- nppiRGBToGray\_8u\_C3C1R, 550
- nppiRGBToHLS\_8u\_AC4R, 550
- nppiRGBToHLS\_8u\_C3R, 551
- nppiRGBToHSV\_8u\_AC4R, 551
- nppiRGBToHSV\_8u\_C3R, 551
- nppiRGBToLUV\_8u\_AC4R, 552
- nppiRGBToLUV\_8u\_C3R, 552
- nppiRGBToXYZ\_8u\_AC4R, 552
- nppiRGBToXYZ\_8u\_C3R, 553
- nppiRGBToYCbCr420\_8u\_C3P3R, 553
- nppiRGBToYCbCr422\_8u\_C3C2R, 553
- nppiRGBToYCbCr422\_8u\_C3P3R, 554
- nppiRGBToYCbCr422\_8u\_P3C2R, 554
- nppiRGBToYCbCr\_8u\_AC4P3R, 555
- nppiRGBToYCbCr\_8u\_AC4R, 555
- nppiRGBToYCbCr\_8u\_C3P3R, 555
- nppiRGBToYCbCr\_8u\_C3R, 556
- nppiRGBToYCbCr\_8u\_P3R, 556
- nppiRGBToYCC\_8u\_AC4R, 556
- nppiRGBToYCC\_8u\_C3R, 557
- nppiRGBToYCrCb420\_8u\_AC4P3R, 557
- nppiRGBToYCrCb422\_8u\_C3C2R, 557
- nppiRGBToYCrCb422\_8u\_P3C2R, 558
- nppiRGBToYUV420\_8u\_C3P3R, 558
- nppiRGBToYUV420\_8u\_P3R, 558
- nppiRGBToYUV422\_8u\_C3C2R, 559
- nppiRGBToYUV422\_8u\_C3P3R, 559
- nppiRGBToYUV422\_8u\_P3R, 559
- nppiRGBToYUV\_8u\_AC4P4R, 560
- nppiRGBToYUV\_8u\_AC4R, 560
- nppiRGBToYUV\_8u\_C3P3R, 561
- nppiRGBToYUV\_8u\_C3R, 561
- nppiRGBToYUV\_8u\_P3R, 561
- nppiXYZToRGB\_8u\_AC4R, 562
- nppiXYZToRGB\_8u\_C3R, 562
- nppiYCbCr411ToBGR\_8u\_P3C3R, 562
- nppiYCbCr411ToBGR\_8u\_P3C4R, 563
- nppiYCbCr420ToBGR\_709CSC\_8u\_P3C3R, 563
- nppiYCbCr420ToBGR\_709HDTV\_8u\_P3C4R, 563
- nppiYCbCr420ToBGR\_8u\_P3C3R, 564
- nppiYCbCr420ToBGR\_8u\_P3C4R, 564
- nppiYCbCr420ToRGB\_8u\_P3C3R, 565
- nppiYCbCr422ToBGR\_8u\_C2C3R, 565
- nppiYCbCr422ToBGR\_8u\_C2C4R, 565
- nppiYCbCr422ToBGR\_8u\_P3C3R, 566
- nppiYCbCr422ToRGB\_8u\_C2C3R, 566
- nppiYCbCr422ToRGB\_8u\_C2P3R, 566
- nppiYCbCr422ToRGB\_8u\_P3C3R, 567
- nppiYCbCrToBGR\_709CSC\_8u\_P3C3R, 567
- nppiYCbCrToBGR\_709CSC\_8u\_P3C4R, 567
- nppiYCbCrToBGR\_8u\_P3C3R, 568
- nppiYCbCrToBGR\_8u\_P3C4R, 568
- nppiYCbCrToRGB\_8u\_AC4R, 569
- nppiYCbCrToRGB\_8u\_C3R, 569
- nppiYCbCrToRGB\_8u\_P3C3R, 569
- nppiYCbCrToRGB\_8u\_P3C4R, 570
- nppiYCbCrToRGB\_8u\_P3R, 570
- nppiYCCToRGB\_8u\_AC4R, 570
- nppiYCCToRGB\_8u\_C3R, 571
- nppiYCrCb420ToRGB\_8u\_P3C4R, 571
- nppiYCrCb422ToRGB\_8u\_C2C3R, 571
- nppiYCrCb422ToRGB\_8u\_C2P3R, 572
- nppiYUV420ToBGR\_8u\_P3C3R, 572
- nppiYUV420ToRGB\_8u\_P3AC4R, 572
- nppiYUV420ToRGB\_8u\_P3C3R, 573
- nppiYUV420ToRGB\_8u\_P3R, 573
- nppiYUV422ToRGB\_8u\_C2C3R, 573
- nppiYUV422ToRGB\_8u\_P3AC4R, 574
- nppiYUV422ToRGB\_8u\_P3C3R, 574
- nppiYUV422ToRGB\_8u\_P3R, 574
- nppiYUVToBGR\_8u\_AC4R, 575
- nppiYUVToBGR\_8u\_C3R, 575
- nppiYUVToBGR\_8u\_P3C3R, 575
- nppiYUVToBGR\_8u\_P3R, 576
- nppiYUVToRGB\_8u\_AC4R, 576
- nppiYUVToRGB\_8u\_C3R, 576
- nppiYUVToRGB\_8u\_P3C3R, 577
- nppiYUVToRGB\_8u\_P3R, 577
- image\_color\_processing
  - nppiColorTwist32f\_16s\_AC4IR, 628
  - nppiColorTwist32f\_16s\_AC4R, 628
  - nppiColorTwist32f\_16s\_C3IR, 628
  - nppiColorTwist32f\_16s\_C3R, 629
  - nppiColorTwist32f\_16s\_IP3R, 629
  - nppiColorTwist32f\_16s\_P3R, 629

nppiColorTwist32f\_16u\_AC4IR, 630  
nppiColorTwist32f\_16u\_AC4R, 630  
nppiColorTwist32f\_16u\_C3IR, 631  
nppiColorTwist32f\_16u\_C3R, 631  
nppiColorTwist32f\_16u\_IP3R, 631  
nppiColorTwist32f\_16u\_P3R, 632  
nppiColorTwist32f\_8s\_AC4IR, 632  
nppiColorTwist32f\_8s\_AC4R, 633  
nppiColorTwist32f\_8s\_C3IR, 633  
nppiColorTwist32f\_8s\_C3R, 633  
nppiColorTwist32f\_8s\_C4IR, 634  
nppiColorTwist32f\_8s\_C4R, 634  
nppiColorTwist32f\_8s\_IP3R, 634  
nppiColorTwist32f\_8s\_P3R, 635  
nppiColorTwist32f\_8u\_AC4IR, 635  
nppiColorTwist32f\_8u\_AC4R, 636  
nppiColorTwist32f\_8u\_C3IR, 636  
nppiColorTwist32f\_8u\_C3R, 636  
nppiColorTwist32f\_8u\_C4IR, 637  
nppiColorTwist32f\_8u\_C4R, 637  
nppiColorTwist32f\_8u\_IP3R, 637  
nppiColorTwist32f\_8u\_P3R, 638  
nppiColorTwist\_32f\_AC4IR, 638  
nppiColorTwist\_32f\_AC4R, 639  
nppiColorTwist\_32f\_C3IR, 639  
nppiColorTwist\_32f\_C3R, 639  
nppiColorTwist\_32f\_C4IR, 640  
nppiColorTwist\_32f\_C4R, 640  
nppiColorTwist\_32f\_IP3R, 640  
nppiColorTwist\_32f\_P3R, 641  
nppiLUT\_16s\_AC4IR, 641  
nppiLUT\_16s\_AC4R, 642  
nppiLUT\_16s\_C1IR, 642  
nppiLUT\_16s\_C1R, 643  
nppiLUT\_16s\_C3IR, 643  
nppiLUT\_16s\_C3R, 644  
nppiLUT\_16s\_C4IR, 644  
nppiLUT\_16s\_C4R, 645  
nppiLUT\_16u\_AC4IR, 645  
nppiLUT\_16u\_AC4R, 646  
nppiLUT\_16u\_C1IR, 646  
nppiLUT\_16u\_C1R, 647  
nppiLUT\_16u\_C3IR, 647  
nppiLUT\_16u\_C3R, 648  
nppiLUT\_16u\_C4IR, 648  
nppiLUT\_16u\_C4R, 649  
nppiLUT\_32f\_AC4IR, 649  
nppiLUT\_32f\_AC4R, 650  
nppiLUT\_32f\_C1IR, 650  
nppiLUT\_32f\_C1R, 651  
nppiLUT\_32f\_C3IR, 651  
nppiLUT\_32f\_C3R, 652  
nppiLUT\_32f\_C4IR, 652  
nppiLUT\_32f\_C4R, 653  
nppiLUT\_8u\_AC4IR, 653  
nppiLUT\_8u\_AC4R, 654  
nppiLUT\_8u\_C1IR, 654  
nppiLUT\_8u\_C1R, 655  
nppiLUT\_8u\_C3IR, 655  
nppiLUT\_8u\_C3R, 656  
nppiLUT\_8u\_C4IR, 656  
nppiLUT\_8u\_C4R, 657  
nppiLUT\_Cubic\_16s\_AC4IR, 657  
nppiLUT\_Cubic\_16s\_AC4R, 658  
nppiLUT\_Cubic\_16s\_C1IR, 658  
nppiLUT\_Cubic\_16s\_C1R, 659  
nppiLUT\_Cubic\_16s\_C3IR, 659  
nppiLUT\_Cubic\_16s\_C3R, 660  
nppiLUT\_Cubic\_16s\_C4IR, 660  
nppiLUT\_Cubic\_16s\_C4R, 661  
nppiLUT\_Cubic\_16u\_AC4IR, 661  
nppiLUT\_Cubic\_16u\_AC4R, 662  
nppiLUT\_Cubic\_16u\_C1IR, 662  
nppiLUT\_Cubic\_16u\_C1R, 663  
nppiLUT\_Cubic\_16u\_C3IR, 663  
nppiLUT\_Cubic\_16u\_C3R, 664  
nppiLUT\_Cubic\_16u\_C4IR, 664  
nppiLUT\_Cubic\_16u\_C4R, 665  
nppiLUT\_Cubic\_32f\_AC4IR, 665  
nppiLUT\_Cubic\_32f\_AC4R, 666  
nppiLUT\_Cubic\_32f\_C1IR, 666  
nppiLUT\_Cubic\_32f\_C1R, 667  
nppiLUT\_Cubic\_32f\_C3IR, 667  
nppiLUT\_Cubic\_32f\_C3R, 668  
nppiLUT\_Cubic\_32f\_C4IR, 668  
nppiLUT\_Cubic\_32f\_C4R, 669  
nppiLUT\_Cubic\_8u\_AC4IR, 669  
nppiLUT\_Cubic\_8u\_AC4R, 670  
nppiLUT\_Cubic\_8u\_C1IR, 670  
nppiLUT\_Cubic\_8u\_C1R, 671  
nppiLUT\_Cubic\_8u\_C3IR, 671  
nppiLUT\_Cubic\_8u\_C3R, 672  
nppiLUT\_Cubic\_8u\_C4IR, 672  
nppiLUT\_Cubic\_8u\_C4R, 673  
nppiLUT\_Linear\_16s\_AC4IR, 673  
nppiLUT\_Linear\_16s\_AC4R, 674  
nppiLUT\_Linear\_16s\_C1IR, 675  
nppiLUT\_Linear\_16s\_C1R, 675  
nppiLUT\_Linear\_16s\_C3IR, 675  
nppiLUT\_Linear\_16s\_C3R, 676  
nppiLUT\_Linear\_16s\_C4IR, 676  
nppiLUT\_Linear\_16s\_C4R, 677  
nppiLUT\_Linear\_16u\_AC4IR, 677  
nppiLUT\_Linear\_16u\_AC4R, 678  
nppiLUT\_Linear\_16u\_C1IR, 679  
nppiLUT\_Linear\_16u\_C1R, 679  
nppiLUT\_Linear\_16u\_C3IR, 679  
nppiLUT\_Linear\_16u\_C3R, 680

- nppiLUT\_Linear\_16u\_C4IR, [680](#)
- nppiLUT\_Linear\_16u\_C4R, [681](#)
- nppiLUT\_Linear\_32f\_AC4IR, [681](#)
- nppiLUT\_Linear\_32f\_AC4R, [682](#)
- nppiLUT\_Linear\_32f\_C1IR, [682](#)
- nppiLUT\_Linear\_32f\_C1R, [683](#)
- nppiLUT\_Linear\_32f\_C3IR, [683](#)
- nppiLUT\_Linear\_32f\_C3R, [684](#)
- nppiLUT\_Linear\_32f\_C4IR, [684](#)
- nppiLUT\_Linear\_32f\_C4R, [685](#)
- nppiLUT\_Linear\_8u\_AC4IR, [685](#)
- nppiLUT\_Linear\_8u\_AC4R, [686](#)
- nppiLUT\_Linear\_8u\_C1IR, [687](#)
- nppiLUT\_Linear\_8u\_C1R, [687](#)
- nppiLUT\_Linear\_8u\_C3IR, [688](#)
- nppiLUT\_Linear\_8u\_C3R, [688](#)
- nppiLUT\_Linear\_8u\_C4IR, [689](#)
- nppiLUT\_Linear\_8u\_C4R, [689](#)
- nppiLUT\_Trilinear\_8u\_AC4IR, [690](#)
- nppiLUT\_Trilinear\_8u\_AC4R, [691](#)
- nppiLUT\_Trilinear\_8u\_C4R, [691](#)
- nppiLUTPalette\_16u24u\_C1R, [692](#)
- nppiLUTPalette\_16u32u\_C1R, [692](#)
- nppiLUTPalette\_16u8u\_C1R, [693](#)
- nppiLUTPalette\_16u\_AC4R, [693](#)
- nppiLUTPalette\_16u\_C1R, [694](#)
- nppiLUTPalette\_16u\_C3R, [694](#)
- nppiLUTPalette\_16u\_C4R, [695](#)
- nppiLUTPalette\_8u24u\_C1R, [695](#)
- nppiLUTPalette\_8u32u\_C1R, [696](#)
- nppiLUTPalette\_8u\_AC4R, [696](#)
- nppiLUTPalette\_8u\_C1R, [697](#)
- nppiLUTPalette\_8u\_C3R, [697](#)
- nppiLUTPalette\_8u\_C4R, [698](#)
- nppiLUTPaletteSwap\_16u\_C3A0C4R, [698](#)
- nppiLUTPaletteSwap\_8u\_C3A0C4R, [699](#)
- image\_color\_sampling\_format\_conversion
  - nppiCbYCr422ToYCbCr411\_8u\_C2P3R, [585](#)
  - nppiCbYCr422ToYCbCr420\_8u\_C2P2R, [586](#)
  - nppiCbYCr422ToYCbCr420\_8u\_C2P3R, [586](#)
  - nppiCbYCr422ToYCbCr422\_8u\_C2P3R, [586](#)
  - nppiCbYCr422ToYCbCr422\_8u\_C2R, [587](#)
  - nppiCbYCr422ToYCrCb420\_8u\_C2P3R, [587](#)
  - nppiYCbCr411\_8u\_P2P3R, [588](#)
  - nppiYCbCr411\_8u\_P3P2R, [588](#)
  - nppiYCbCr411ToYCbCr420\_8u\_P2P3R, [588](#)
  - nppiYCbCr411ToYCbCr420\_8u\_P3P2R, [589](#)
  - nppiYCbCr411ToYCbCr420\_8u\_P3R, [589](#)
  - nppiYCbCr411ToYCbCr422\_8u\_P2C2R, [590](#)
  - nppiYCbCr411ToYCbCr422\_8u\_P2P3R, [590](#)
  - nppiYCbCr411ToYCbCr422\_8u\_P3C2R, [590](#)
  - nppiYCbCr411ToYCbCr422\_8u\_P3R, [591](#)
  - nppiYCbCr411ToYCrCb420\_8u\_P2P3R, [591](#)
  - nppiYCbCr411ToYCrCb422\_8u\_P3C2R, [592](#)
  - nppiYCbCr411ToYCrCb422\_8u\_P3R, [592](#)
  - nppiYCbCr420\_8u\_P2P3R, [592](#)
  - nppiYCbCr420\_8u\_P3P2R, [593](#)
  - nppiYCbCr420ToCbYCr422\_8u\_P2C2R, [593](#)
  - nppiYCbCr420ToYCbCr411\_8u\_P2P3R, [594](#)
  - nppiYCbCr420ToYCbCr411\_8u\_P3P2R, [594](#)
  - nppiYCbCr420ToYCbCr422\_8u\_P2C2R, [595](#)
  - nppiYCbCr420ToYCbCr422\_8u\_P2P3R, [595](#)
  - nppiYCbCr420ToYCbCr422\_8u\_P3R, [595](#)
  - nppiYCbCr420ToYCrCb420\_8u\_P2P3R, [596](#)
  - nppiYCbCr422\_8u\_C2P3R, [596](#)
  - nppiYCbCr422\_8u\_P3C2R, [597](#)
  - nppiYCbCr422ToCbYCr422\_8u\_C2R, [597](#)
  - nppiYCbCr422ToYCbCr411\_8u\_C2P2R, [597](#)
  - nppiYCbCr422ToYCbCr411\_8u\_C2P3R, [598](#)
  - nppiYCbCr422ToYCbCr411\_8u\_P3P2R, [598](#)
  - nppiYCbCr422ToYCbCr411\_8u\_P3R, [599](#)
  - nppiYCbCr422ToYCbCr420\_8u\_C2P2R, [599](#)
  - nppiYCbCr422ToYCbCr420\_8u\_C2P3R, [600](#)
  - nppiYCbCr422ToYCbCr420\_8u\_P3P2R, [600](#)
  - nppiYCbCr422ToYCbCr420\_8u\_P3R, [600](#)
  - nppiYCbCr422ToYCrCb420\_8u\_C2P3R, [601](#)
  - nppiYCbCr422ToYCrCb422\_8u\_C2R, [601](#)
  - nppiYCbCr422ToYCrCb422\_8u\_P3C2R, [602](#)
  - nppiYCrCb420ToCbYCr422\_8u\_P3C2R, [602](#)
  - nppiYCrCb420ToYCbCr411\_8u\_P3P2R, [602](#)
  - nppiYCrCb420ToYCbCr420\_8u\_P3P2R, [603](#)
  - nppiYCrCb420ToYCbCr422\_8u\_P3C2R, [603](#)
  - nppiYCrCb420ToYCbCr422\_8u\_P3R, [604](#)
  - nppiYCrCb422ToYCbCr411\_8u\_C2P3R, [604](#)
  - nppiYCrCb422ToYCbCr420\_8u\_C2P3R, [605](#)
  - nppiYCrCb422ToYCbCr422\_8u\_C2P3R, [605](#)
- image\_compare\_operations
  - nppiCompare\_16s\_AC4R, [2223](#)
  - nppiCompare\_16s\_C1R, [2224](#)
  - nppiCompare\_16s\_C3R, [2224](#)
  - nppiCompare\_16s\_C4R, [2224](#)
  - nppiCompare\_16u\_AC4R, [2225](#)
  - nppiCompare\_16u\_C1R, [2225](#)
  - nppiCompare\_16u\_C3R, [2226](#)
  - nppiCompare\_16u\_C4R, [2226](#)
  - nppiCompare\_32f\_AC4R, [2227](#)
  - nppiCompare\_32f\_C1R, [2227](#)
  - nppiCompare\_32f\_C3R, [2228](#)
  - nppiCompare\_32f\_C4R, [2228](#)
  - nppiCompare\_8u\_AC4R, [2229](#)
  - nppiCompare\_8u\_C1R, [2229](#)
  - nppiCompare\_8u\_C3R, [2230](#)
  - nppiCompare\_8u\_C4R, [2230](#)
  - nppiCompareC\_16s\_AC4R, [2231](#)
  - nppiCompareC\_16s\_C1R, [2231](#)
  - nppiCompareC\_16s\_C3R, [2232](#)
  - nppiCompareC\_16s\_C4R, [2232](#)
  - nppiCompareC\_16u\_AC4R, [2233](#)

- [nppiCompareC\\_16u\\_C1R](#), [2233](#)
- [nppiCompareC\\_16u\\_C3R](#), [2233](#)
- [nppiCompareC\\_16u\\_C4R](#), [2234](#)
- [nppiCompareC\\_32f\\_AC4R](#), [2234](#)
- [nppiCompareC\\_32f\\_C1R](#), [2235](#)
- [nppiCompareC\\_32f\\_C3R](#), [2235](#)
- [nppiCompareC\\_32f\\_C4R](#), [2236](#)
- [nppiCompareC\\_8u\\_AC4R](#), [2236](#)
- [nppiCompareC\\_8u\\_C1R](#), [2236](#)
- [nppiCompareC\\_8u\\_C3R](#), [2237](#)
- [nppiCompareC\\_8u\\_C4R](#), [2237](#)
- [nppiCompareEqualEps\\_32f\\_AC4R](#), [2238](#)
- [nppiCompareEqualEps\\_32f\\_C1R](#), [2238](#)
- [nppiCompareEqualEps\\_32f\\_C3R](#), [2239](#)
- [nppiCompareEqualEps\\_32f\\_C4R](#), [2239](#)
- [nppiCompareEqualEpsC\\_32f\\_AC4R](#), [2240](#)
- [nppiCompareEqualEpsC\\_32f\\_C1R](#), [2240](#)
- [nppiCompareEqualEpsC\\_32f\\_C3R](#), [2241](#)
- [nppiCompareEqualEpsC\\_32f\\_C4R](#), [2241](#)
- [image\\_complement\\_color\\_key](#)
  - [nppiAlphaCompColorKey\\_8u\\_AC4R](#), [612](#)
  - [nppiCompColorKey\\_8u\\_C1R](#), [613](#)
  - [nppiCompColorKey\\_8u\\_C3R](#), [613](#)
  - [nppiCompColorKey\\_8u\\_C4R](#), [614](#)
- [image\\_compression](#)
  - [nppiDecodeHuffmanScanHost\\_JPEG\\_-8u16s\\_P1R](#), [702](#)
  - [nppiDecodeHuffmanScanHost\\_JPEG\\_-8u16s\\_P3R](#), [702](#)
  - [NppiDecodeHuffmanSpec](#), [702](#)
  - [nppiDecodeHuffmanSpecFreeHost\\_JPEG](#), [703](#)
  - [nppiDecodeHuffmanSpecGetBufSize\\_JPEG](#), [703](#)
  - [nppiDecodeHuffmanSpecInitAllocHost\\_-JPEG](#), [703](#)
  - [nppiDecodeHuffmanSpecInitHost\\_JPEG](#), [704](#)
- [image\\_convert](#)
  - [nppiConvert\\_16s16u\\_C1Rs](#), [809](#)
  - [nppiConvert\\_16s32f\\_AC4R](#), [809](#)
  - [nppiConvert\\_16s32f\\_C1R](#), [810](#)
  - [nppiConvert\\_16s32f\\_C3R](#), [810](#)
  - [nppiConvert\\_16s32f\\_C4R](#), [810](#)
  - [nppiConvert\\_16s32s\\_AC4R](#), [811](#)
  - [nppiConvert\\_16s32s\\_C1R](#), [811](#)
  - [nppiConvert\\_16s32s\\_C3R](#), [811](#)
  - [nppiConvert\\_16s32s\\_C4R](#), [812](#)
  - [nppiConvert\\_16s32u\\_C1Rs](#), [812](#)
  - [nppiConvert\\_16s8s\\_C1RSfs](#), [812](#)
  - [nppiConvert\\_16s8u\\_AC4R](#), [813](#)
  - [nppiConvert\\_16s8u\\_C1R](#), [813](#)
  - [nppiConvert\\_16s8u\\_C3R](#), [813](#)
  - [nppiConvert\\_16s8u\\_C4R](#), [814](#)
  - [nppiConvert\\_16u16s\\_C1RSfs](#), [814](#)
  - [nppiConvert\\_16u32f\\_AC4R](#), [814](#)
  - [nppiConvert\\_16u32f\\_C1R](#), [815](#)
  - [nppiConvert\\_16u32f\\_C3R](#), [815](#)
  - [nppiConvert\\_16u32f\\_C4R](#), [815](#)
  - [nppiConvert\\_16u32s\\_AC4R](#), [816](#)
  - [nppiConvert\\_16u32s\\_C1R](#), [816](#)
  - [nppiConvert\\_16u32s\\_C3R](#), [816](#)
  - [nppiConvert\\_16u32s\\_C4R](#), [817](#)
  - [nppiConvert\\_16u32u\\_C1R](#), [817](#)
  - [nppiConvert\\_16u8s\\_C1RSfs](#), [817](#)
  - [nppiConvert\\_16u8u\\_AC4R](#), [818](#)
  - [nppiConvert\\_16u8u\\_C1R](#), [818](#)
  - [nppiConvert\\_16u8u\\_C3R](#), [818](#)
  - [nppiConvert\\_16u8u\\_C4R](#), [819](#)
  - [nppiConvert\\_32f16s\\_AC4R](#), [819](#)
  - [nppiConvert\\_32f16s\\_C1R](#), [819](#)
  - [nppiConvert\\_32f16s\\_C1RSfs](#), [820](#)
  - [nppiConvert\\_32f16s\\_C3R](#), [820](#)
  - [nppiConvert\\_32f16s\\_C4R](#), [821](#)
  - [nppiConvert\\_32f16u\\_AC4R](#), [821](#)
  - [nppiConvert\\_32f16u\\_C1R](#), [821](#)
  - [nppiConvert\\_32f16u\\_C1RSfs](#), [822](#)
  - [nppiConvert\\_32f16u\\_C3R](#), [822](#)
  - [nppiConvert\\_32f16u\\_C4R](#), [823](#)
  - [nppiConvert\\_32f32s\\_C1RSfs](#), [823](#)
  - [nppiConvert\\_32f32u\\_C1RSfs](#), [823](#)
  - [nppiConvert\\_32f8s\\_AC4R](#), [824](#)
  - [nppiConvert\\_32f8s\\_C1R](#), [824](#)
  - [nppiConvert\\_32f8s\\_C1RSfs](#), [825](#)
  - [nppiConvert\\_32f8s\\_C3R](#), [825](#)
  - [nppiConvert\\_32f8s\\_C4R](#), [825](#)
  - [nppiConvert\\_32f8u\\_AC4R](#), [826](#)
  - [nppiConvert\\_32f8u\\_C1R](#), [826](#)
  - [nppiConvert\\_32f8u\\_C1RSfs](#), [826](#)
  - [nppiConvert\\_32f8u\\_C3R](#), [827](#)
  - [nppiConvert\\_32f8u\\_C4R](#), [827](#)
  - [nppiConvert\\_32s16s\\_C1RSfs](#), [828](#)
  - [nppiConvert\\_32s16u\\_C1RSfs](#), [828](#)
  - [nppiConvert\\_32s32f\\_C1R](#), [828](#)
  - [nppiConvert\\_32s32u\\_C1Rs](#), [829](#)
  - [nppiConvert\\_32s8s\\_AC4R](#), [829](#)
  - [nppiConvert\\_32s8s\\_C1R](#), [829](#)
  - [nppiConvert\\_32s8s\\_C3R](#), [830](#)
  - [nppiConvert\\_32s8s\\_C4R](#), [830](#)
  - [nppiConvert\\_32s8u\\_AC4R](#), [830](#)
  - [nppiConvert\\_32s8u\\_C1R](#), [831](#)
  - [nppiConvert\\_32s8u\\_C3R](#), [831](#)
  - [nppiConvert\\_32s8u\\_C4R](#), [831](#)
  - [nppiConvert\\_32u16s\\_C1RSfs](#), [832](#)
  - [nppiConvert\\_32u16u\\_C1RSfs](#), [832](#)
  - [nppiConvert\\_32u32f\\_C1R](#), [833](#)
  - [nppiConvert\\_32u32s\\_C1RSfs](#), [833](#)
  - [nppiConvert\\_32u8s\\_C1RSfs](#), [833](#)
  - [nppiConvert\\_32u8u\\_C1RSfs](#), [834](#)



- nppiConvert\_8s16s\_C1R, [834](#)
- nppiConvert\_8s16u\_C1Rs, [835](#)
- nppiConvert\_8s32f\_AC4R, [835](#)
- nppiConvert\_8s32f\_C1R, [835](#)
- nppiConvert\_8s32f\_C3R, [836](#)
- nppiConvert\_8s32f\_C4R, [836](#)
- nppiConvert\_8s32s\_AC4R, [836](#)
- nppiConvert\_8s32s\_C1R, [837](#)
- nppiConvert\_8s32s\_C3R, [837](#)
- nppiConvert\_8s32s\_C4R, [837](#)
- nppiConvert\_8s32u\_C1Rs, [838](#)
- nppiConvert\_8s8u\_C1Rs, [838](#)
- nppiConvert\_8u16s\_AC4R, [838](#)
- nppiConvert\_8u16s\_C1R, [839](#)
- nppiConvert\_8u16s\_C3R, [839](#)
- nppiConvert\_8u16s\_C4R, [839](#)
- nppiConvert\_8u16u\_AC4R, [840](#)
- nppiConvert\_8u16u\_C1R, [840](#)
- nppiConvert\_8u16u\_C3R, [840](#)
- nppiConvert\_8u16u\_C4R, [841](#)
- nppiConvert\_8u32f\_AC4R, [841](#)
- nppiConvert\_8u32f\_C1R, [841](#)
- nppiConvert\_8u32f\_C3R, [842](#)
- nppiConvert\_8u32f\_C4R, [842](#)
- nppiConvert\_8u32s\_AC4R, [842](#)
- nppiConvert\_8u32s\_C1R, [843](#)
- nppiConvert\_8u32s\_C3R, [843](#)
- nppiConvert\_8u32s\_C4R, [843](#)
- nppiConvert\_8u8s\_C1RSfs, [844](#)
- image\_convolution
  - nppiFilter32f\_16s\_AC4R, [1066](#)
  - nppiFilter32f\_16s\_C1R, [1066](#)
  - nppiFilter32f\_16s\_C3R, [1066](#)
  - nppiFilter32f\_16s\_C4R, [1067](#)
  - nppiFilter32f\_16u\_AC4R, [1067](#)
  - nppiFilter32f\_16u\_C1R, [1068](#)
  - nppiFilter32f\_16u\_C3R, [1068](#)
  - nppiFilter32f\_16u\_C4R, [1069](#)
  - nppiFilter32f\_32s\_AC4R, [1069](#)
  - nppiFilter32f\_32s\_C1R, [1070](#)
  - nppiFilter32f\_32s\_C3R, [1070](#)
  - nppiFilter32f\_32s\_C4R, [1071](#)
  - nppiFilter32f\_8s16s\_AC4R, [1071](#)
  - nppiFilter32f\_8s16s\_C1R, [1072](#)
  - nppiFilter32f\_8s16s\_C3R, [1072](#)
  - nppiFilter32f\_8s16s\_C4R, [1073](#)
  - nppiFilter32f\_8s\_AC4R, [1073](#)
  - nppiFilter32f\_8s\_C1R, [1074](#)
  - nppiFilter32f\_8s\_C3R, [1074](#)
  - nppiFilter32f\_8s\_C4R, [1075](#)
  - nppiFilter32f\_8u16s\_AC4R, [1075](#)
  - nppiFilter32f\_8u16s\_C1R, [1076](#)
  - nppiFilter32f\_8u16s\_C3R, [1076](#)
  - nppiFilter32f\_8u16s\_C4R, [1077](#)
  - nppiFilter32f\_8u\_AC4R, [1077](#)
  - nppiFilter32f\_8u\_C1R, [1078](#)
  - nppiFilter32f\_8u\_C3R, [1078](#)
  - nppiFilter32f\_8u\_C4R, [1079](#)
  - nppiFilter\_16s\_AC4R, [1079](#)
  - nppiFilter\_16s\_C1R, [1080](#)
  - nppiFilter\_16s\_C3R, [1080](#)
  - nppiFilter\_16s\_C4R, [1081](#)
  - nppiFilter\_16u\_AC4R, [1081](#)
  - nppiFilter\_16u\_C1R, [1082](#)
  - nppiFilter\_16u\_C3R, [1082](#)
  - nppiFilter\_16u\_C4R, [1083](#)
  - nppiFilter\_32f\_AC4R, [1083](#)
  - nppiFilter\_32f\_C1R, [1084](#)
  - nppiFilter\_32f\_C3R, [1084](#)
  - nppiFilter\_32f\_C4R, [1085](#)
  - nppiFilter\_64f\_C1R, [1085](#)
  - nppiFilter\_8u\_AC4R, [1086](#)
  - nppiFilter\_8u\_C1R, [1086](#)
  - nppiFilter\_8u\_C3R, [1087](#)
  - nppiFilter\_8u\_C4R, [1087](#)
  - nppiFilterBorder32f\_16s\_AC4R, [1088](#)
  - nppiFilterBorder32f\_16s\_C1R, [1088](#)
  - nppiFilterBorder32f\_16s\_C3R, [1089](#)
  - nppiFilterBorder32f\_16s\_C4R, [1089](#)
  - nppiFilterBorder32f\_16u\_AC4R, [1090](#)
  - nppiFilterBorder32f\_16u\_C1R, [1090](#)
  - nppiFilterBorder32f\_16u\_C3R, [1091](#)
  - nppiFilterBorder32f\_16u\_C4R, [1091](#)
  - nppiFilterBorder32f\_32s\_AC4R, [1092](#)
  - nppiFilterBorder32f\_32s\_C1R, [1092](#)
  - nppiFilterBorder32f\_32s\_C3R, [1093](#)
  - nppiFilterBorder32f\_32s\_C4R, [1093](#)
  - nppiFilterBorder32f\_8s16s\_AC4R, [1094](#)
  - nppiFilterBorder32f\_8s16s\_C1R, [1094](#)
  - nppiFilterBorder32f\_8s16s\_C3R, [1095](#)
  - nppiFilterBorder32f\_8s16s\_C4R, [1095](#)
  - nppiFilterBorder32f\_8s\_AC4R, [1096](#)
  - nppiFilterBorder32f\_8s\_C1R, [1096](#)
  - nppiFilterBorder32f\_8s\_C3R, [1097](#)
  - nppiFilterBorder32f\_8s\_C4R, [1097](#)
  - nppiFilterBorder32f\_8u16s\_AC4R, [1098](#)
  - nppiFilterBorder32f\_8u16s\_C1R, [1098](#)
  - nppiFilterBorder32f\_8u16s\_C3R, [1099](#)
  - nppiFilterBorder32f\_8u16s\_C4R, [1099](#)
  - nppiFilterBorder32f\_8u\_AC4R, [1100](#)
  - nppiFilterBorder32f\_8u\_C1R, [1100](#)
  - nppiFilterBorder32f\_8u\_C3R, [1101](#)
  - nppiFilterBorder32f\_8u\_C4R, [1101](#)
  - nppiFilterBorder\_16s\_AC4R, [1102](#)
  - nppiFilterBorder\_16s\_C1R, [1103](#)
  - nppiFilterBorder\_16s\_C3R, [1103](#)
  - nppiFilterBorder\_16s\_C4R, [1104](#)
  - nppiFilterBorder\_16u\_AC4R, [1104](#)

- nppiFilterBorder\_16u\_C1R, [1105](#)
- nppiFilterBorder\_16u\_C3R, [1106](#)
- nppiFilterBorder\_16u\_C4R, [1106](#)
- nppiFilterBorder\_32f\_AC4R, [1107](#)
- nppiFilterBorder\_32f\_C1R, [1107](#)
- nppiFilterBorder\_32f\_C3R, [1108](#)
- nppiFilterBorder\_32f\_C4R, [1108](#)
- nppiFilterBorder\_8u\_AC4R, [1109](#)
- nppiFilterBorder\_8u\_C1R, [1110](#)
- nppiFilterBorder\_8u\_C3R, [1110](#)
- nppiFilterBorder\_8u\_C4R, [1111](#)
- image\_copy
  - nppiCopy\_16s\_AC4MR, [763](#)
  - nppiCopy\_16s\_AC4R, [764](#)
  - nppiCopy\_16s\_C1C3R, [764](#)
  - nppiCopy\_16s\_C1C4R, [764](#)
  - nppiCopy\_16s\_C1MR, [765](#)
  - nppiCopy\_16s\_C1R, [765](#)
  - nppiCopy\_16s\_C3C1R, [765](#)
  - nppiCopy\_16s\_C3CR, [766](#)
  - nppiCopy\_16s\_C3MR, [766](#)
  - nppiCopy\_16s\_C3P3R, [766](#)
  - nppiCopy\_16s\_C3R, [767](#)
  - nppiCopy\_16s\_C4C1R, [767](#)
  - nppiCopy\_16s\_C4CR, [767](#)
  - nppiCopy\_16s\_C4MR, [768](#)
  - nppiCopy\_16s\_C4P4R, [768](#)
  - nppiCopy\_16s\_C4R, [768](#)
  - nppiCopy\_16s\_P3C3R, [769](#)
  - nppiCopy\_16s\_P4C4R, [769](#)
  - nppiCopy\_16sc\_AC4R, [769](#)
  - nppiCopy\_16sc\_C1R, [770](#)
  - nppiCopy\_16sc\_C2R, [770](#)
  - nppiCopy\_16sc\_C3R, [770](#)
  - nppiCopy\_16sc\_C4R, [771](#)
  - nppiCopy\_16u\_AC4MR, [771](#)
  - nppiCopy\_16u\_AC4R, [771](#)
  - nppiCopy\_16u\_C1C3R, [772](#)
  - nppiCopy\_16u\_C1C4R, [772](#)
  - nppiCopy\_16u\_C1MR, [772](#)
  - nppiCopy\_16u\_C1R, [773](#)
  - nppiCopy\_16u\_C3C1R, [773](#)
  - nppiCopy\_16u\_C3CR, [773](#)
  - nppiCopy\_16u\_C3MR, [774](#)
  - nppiCopy\_16u\_C3P3R, [774](#)
  - nppiCopy\_16u\_C3R, [774](#)
  - nppiCopy\_16u\_C4C1R, [775](#)
  - nppiCopy\_16u\_C4CR, [775](#)
  - nppiCopy\_16u\_C4MR, [775](#)
  - nppiCopy\_16u\_C4P4R, [776](#)
  - nppiCopy\_16u\_C4R, [776](#)
  - nppiCopy\_16u\_P3C3R, [776](#)
  - nppiCopy\_16u\_P4C4R, [777](#)
  - nppiCopy\_32f\_AC4MR, [777](#)
  - nppiCopy\_32f\_AC4R, [777](#)
  - nppiCopy\_32f\_C1C3R, [778](#)
  - nppiCopy\_32f\_C1C4R, [778](#)
  - nppiCopy\_32f\_C1MR, [778](#)
  - nppiCopy\_32f\_C1R, [779](#)
  - nppiCopy\_32f\_C3C1R, [779](#)
  - nppiCopy\_32f\_C3CR, [779](#)
  - nppiCopy\_32f\_C3MR, [780](#)
  - nppiCopy\_32f\_C3P3R, [780](#)
  - nppiCopy\_32f\_C3R, [780](#)
  - nppiCopy\_32f\_C4C1R, [781](#)
  - nppiCopy\_32f\_C4CR, [781](#)
  - nppiCopy\_32f\_C4MR, [781](#)
  - nppiCopy\_32f\_C4P4R, [782](#)
  - nppiCopy\_32f\_C4R, [782](#)
  - nppiCopy\_32f\_P3C3R, [782](#)
  - nppiCopy\_32f\_P4C4R, [783](#)
  - nppiCopy\_32fc\_AC4R, [783](#)
  - nppiCopy\_32fc\_C1R, [783](#)
  - nppiCopy\_32fc\_C2R, [784](#)
  - nppiCopy\_32fc\_C3R, [784](#)
  - nppiCopy\_32fc\_C4R, [784](#)
  - nppiCopy\_32s\_AC4MR, [785](#)
  - nppiCopy\_32s\_AC4R, [785](#)
  - nppiCopy\_32s\_C1C3R, [785](#)
  - nppiCopy\_32s\_C1C4R, [786](#)
  - nppiCopy\_32s\_C1MR, [786](#)
  - nppiCopy\_32s\_C1R, [786](#)
  - nppiCopy\_32s\_C3C1R, [787](#)
  - nppiCopy\_32s\_C3CR, [787](#)
  - nppiCopy\_32s\_C3MR, [787](#)
  - nppiCopy\_32s\_C3P3R, [788](#)
  - nppiCopy\_32s\_C3R, [788](#)
  - nppiCopy\_32s\_C4C1R, [788](#)
  - nppiCopy\_32s\_C4CR, [789](#)
  - nppiCopy\_32s\_C4MR, [789](#)
  - nppiCopy\_32s\_C4P4R, [789](#)
  - nppiCopy\_32s\_C4R, [790](#)
  - nppiCopy\_32s\_P3C3R, [790](#)
  - nppiCopy\_32s\_P4C4R, [790](#)
  - nppiCopy\_32sc\_AC4R, [791](#)
  - nppiCopy\_32sc\_C1R, [791](#)
  - nppiCopy\_32sc\_C2R, [791](#)
  - nppiCopy\_32sc\_C3R, [792](#)
  - nppiCopy\_32sc\_C4R, [792](#)
  - nppiCopy\_8s\_AC4R, [792](#)
  - nppiCopy\_8s\_C1R, [793](#)
  - nppiCopy\_8s\_C2R, [793](#)
  - nppiCopy\_8s\_C3R, [793](#)
  - nppiCopy\_8s\_C4R, [794](#)
  - nppiCopy\_8u\_AC4MR, [794](#)
  - nppiCopy\_8u\_AC4R, [794](#)
  - nppiCopy\_8u\_C1C3R, [795](#)
  - nppiCopy\_8u\_C1C4R, [795](#)

- nppiCopy\_8u\_C1MR, 795
- nppiCopy\_8u\_C1R, 796
- nppiCopy\_8u\_C3C1R, 796
- nppiCopy\_8u\_C3CR, 796
- nppiCopy\_8u\_C3MR, 797
- nppiCopy\_8u\_C3P3R, 797
- nppiCopy\_8u\_C3R, 797
- nppiCopy\_8u\_C4C1R, 798
- nppiCopy\_8u\_C4CR, 798
- nppiCopy\_8u\_C4MR, 798
- nppiCopy\_8u\_C4P4R, 799
- nppiCopy\_8u\_C4R, 799
- nppiCopy\_8u\_P3C3R, 799
- nppiCopy\_8u\_P4C4R, 800
- image\_copy\_constant\_border
  - nppiCopyConstBorder\_16s\_AC4R, 862
  - nppiCopyConstBorder\_16s\_C1R, 862
  - nppiCopyConstBorder\_16s\_C3R, 863
  - nppiCopyConstBorder\_16s\_C4R, 863
  - nppiCopyConstBorder\_16u\_AC4R, 864
  - nppiCopyConstBorder\_16u\_C1R, 864
  - nppiCopyConstBorder\_16u\_C3R, 865
  - nppiCopyConstBorder\_16u\_C4R, 865
  - nppiCopyConstBorder\_32f\_AC4R, 866
  - nppiCopyConstBorder\_32f\_C1R, 866
  - nppiCopyConstBorder\_32f\_C3R, 867
  - nppiCopyConstBorder\_32f\_C4R, 867
  - nppiCopyConstBorder\_32s\_AC4R, 868
  - nppiCopyConstBorder\_32s\_C1R, 868
  - nppiCopyConstBorder\_32s\_C3R, 869
  - nppiCopyConstBorder\_32s\_C4R, 869
  - nppiCopyConstBorder\_8u\_AC4R, 870
  - nppiCopyConstBorder\_8u\_C1R, 870
  - nppiCopyConstBorder\_8u\_C3R, 871
  - nppiCopyConstBorder\_8u\_C4R, 871
- image\_copy\_replicate\_border
  - nppiCopyReplicateBorder\_16s\_AC4R, 875
  - nppiCopyReplicateBorder\_16s\_C1R, 875
  - nppiCopyReplicateBorder\_16s\_C3R, 876
  - nppiCopyReplicateBorder\_16s\_C4R, 876
  - nppiCopyReplicateBorder\_16u\_AC4R, 877
  - nppiCopyReplicateBorder\_16u\_C1R, 877
  - nppiCopyReplicateBorder\_16u\_C3R, 878
  - nppiCopyReplicateBorder\_16u\_C4R, 878
  - nppiCopyReplicateBorder\_32f\_AC4R, 878
  - nppiCopyReplicateBorder\_32f\_C1R, 879
  - nppiCopyReplicateBorder\_32f\_C3R, 879
  - nppiCopyReplicateBorder\_32f\_C4R, 880
  - nppiCopyReplicateBorder\_32s\_AC4R, 880
  - nppiCopyReplicateBorder\_32s\_C1R, 881
  - nppiCopyReplicateBorder\_32s\_C3R, 881
  - nppiCopyReplicateBorder\_32s\_C4R, 882
  - nppiCopyReplicateBorder\_8u\_AC4R, 882
  - nppiCopyReplicateBorder\_8u\_C1R, 883
- nppiCopyReplicateBorder\_8u\_C3R, 883
- nppiCopyReplicateBorder\_8u\_C4R, 884
- image\_copy\_sub\_pixel
  - nppiCopySubpix\_16s\_AC4R, 899
  - nppiCopySubpix\_16s\_C1R, 900
  - nppiCopySubpix\_16s\_C3R, 900
  - nppiCopySubpix\_16s\_C4R, 901
  - nppiCopySubpix\_16u\_AC4R, 901
  - nppiCopySubpix\_16u\_C1R, 901
  - nppiCopySubpix\_16u\_C3R, 902
  - nppiCopySubpix\_16u\_C4R, 902
  - nppiCopySubpix\_32f\_AC4R, 903
  - nppiCopySubpix\_32f\_C1R, 903
  - nppiCopySubpix\_32f\_C3R, 903
  - nppiCopySubpix\_32f\_C4R, 904
  - nppiCopySubpix\_32s\_AC4R, 904
  - nppiCopySubpix\_32s\_C1R, 905
  - nppiCopySubpix\_32s\_C3R, 905
  - nppiCopySubpix\_32s\_C4R, 906
  - nppiCopySubpix\_8u\_AC4R, 906
  - nppiCopySubpix\_8u\_C1R, 906
  - nppiCopySubpix\_8u\_C3R, 907
  - nppiCopySubpix\_8u\_C4R, 907
- image\_copy\_wrap\_border
  - nppiCopyWrapBorder\_16s\_AC4R, 887
  - nppiCopyWrapBorder\_16s\_C1R, 887
  - nppiCopyWrapBorder\_16s\_C3R, 888
  - nppiCopyWrapBorder\_16s\_C4R, 888
  - nppiCopyWrapBorder\_16u\_AC4R, 889
  - nppiCopyWrapBorder\_16u\_C1R, 889
  - nppiCopyWrapBorder\_16u\_C3R, 890
  - nppiCopyWrapBorder\_16u\_C4R, 890
  - nppiCopyWrapBorder\_32f\_AC4R, 891
  - nppiCopyWrapBorder\_32f\_C1R, 891
  - nppiCopyWrapBorder\_32f\_C3R, 892
  - nppiCopyWrapBorder\_32f\_C4R, 892
  - nppiCopyWrapBorder\_32s\_AC4R, 893
  - nppiCopyWrapBorder\_32s\_C1R, 893
  - nppiCopyWrapBorder\_32s\_C3R, 894
  - nppiCopyWrapBorder\_32s\_C4R, 894
  - nppiCopyWrapBorder\_8u\_AC4R, 895
  - nppiCopyWrapBorder\_8u\_C1R, 895
  - nppiCopyWrapBorder\_8u\_C3R, 896
  - nppiCopyWrapBorder\_8u\_C4R, 896
- image\_count\_in\_range
  - nppiCountInRange\_32f\_AC4R, 1827
  - nppiCountInRange\_32f\_C1R, 1827
  - nppiCountInRange\_32f\_C3R, 1828
  - nppiCountInRange\_8u\_AC4R, 1828
  - nppiCountInRange\_8u\_C1R, 1829
  - nppiCountInRange\_8u\_C3R, 1829
  - nppiCountInRangeGetBufferHostSize\_32f\_AC4R, 1830



[nppiCountInRangeGetBufferSize\\_32f\\_C1R, 1830](#)  
[nppiCountInRangeGetBufferSize\\_32f\\_C3R, 1830](#)  
[nppiCountInRangeGetBufferSize\\_8u\\_AC4R, 1830](#)  
[nppiCountInRangeGetBufferSize\\_8u\\_C1R, 1831](#)  
[nppiCountInRangeGetBufferSize\\_8u\\_C3R, 1831](#)  
[image\\_dilate](#)  
[nppiDilate\\_16u\\_AC4R, 1368](#)  
[nppiDilate\\_16u\\_C1R, 1368](#)  
[nppiDilate\\_16u\\_C3R, 1369](#)  
[nppiDilate\\_16u\\_C4R, 1369](#)  
[nppiDilate\\_32f\\_AC4R, 1369](#)  
[nppiDilate\\_32f\\_C1R, 1370](#)  
[nppiDilate\\_32f\\_C3R, 1370](#)  
[nppiDilate\\_32f\\_C4R, 1371](#)  
[nppiDilate\\_8u\\_AC4R, 1371](#)  
[nppiDilate\\_8u\\_C1R, 1372](#)  
[nppiDilate\\_8u\\_C3R, 1372](#)  
[nppiDilate\\_8u\\_C4R, 1372](#)  
[image\\_dilate\\_3x3](#)  
[nppiDilate3x3\\_16u\\_AC4R, 1382](#)  
[nppiDilate3x3\\_16u\\_C1R, 1382](#)  
[nppiDilate3x3\\_16u\\_C3R, 1382](#)  
[nppiDilate3x3\\_16u\\_C4R, 1383](#)  
[nppiDilate3x3\\_32f\\_AC4R, 1383](#)  
[nppiDilate3x3\\_32f\\_C1R, 1383](#)  
[nppiDilate3x3\\_32f\\_C3R, 1384](#)  
[nppiDilate3x3\\_32f\\_C4R, 1384](#)  
[nppiDilate3x3\\_64f\\_C1R, 1384](#)  
[nppiDilate3x3\\_8u\\_AC4R, 1385](#)  
[nppiDilate3x3\\_8u\\_C1R, 1385](#)  
[nppiDilate3x3\\_8u\\_C3R, 1385](#)  
[nppiDilate3x3\\_8u\\_C4R, 1386](#)  
[image\\_div](#)  
[nppiDiv\\_16s\\_AC4IRSfs, 281](#)  
[nppiDiv\\_16s\\_AC4RSfs, 281](#)  
[nppiDiv\\_16s\\_C1IRSfs, 282](#)  
[nppiDiv\\_16s\\_C1RSfs, 282](#)  
[nppiDiv\\_16s\\_C3IRSfs, 282](#)  
[nppiDiv\\_16s\\_C3RSfs, 283](#)  
[nppiDiv\\_16s\\_C4IRSfs, 283](#)  
[nppiDiv\\_16s\\_C4RSfs, 284](#)  
[nppiDiv\\_16sc\\_AC4IRSfs, 284](#)  
[nppiDiv\\_16sc\\_AC4RSfs, 284](#)  
[nppiDiv\\_16sc\\_C1IRSfs, 285](#)  
[nppiDiv\\_16sc\\_C1RSfs, 285](#)  
[nppiDiv\\_16sc\\_C3IRSfs, 286](#)  
[nppiDiv\\_16sc\\_C3RSfs, 286](#)  
[nppiDiv\\_16u\\_AC4IRSfs, 287](#)  
[nppiDiv\\_16u\\_AC4RSfs, 287](#)  
[nppiDiv\\_16u\\_C1IRSfs, 287](#)  
[nppiDiv\\_16u\\_C1RSfs, 288](#)  
[nppiDiv\\_16u\\_C3IRSfs, 288](#)  
[nppiDiv\\_16u\\_C3RSfs, 289](#)  
[nppiDiv\\_16u\\_C4IRSfs, 289](#)  
[nppiDiv\\_16u\\_C4RSfs, 289](#)  
[nppiDiv\\_32f\\_AC4IR, 290](#)  
[nppiDiv\\_32f\\_AC4R, 290](#)  
[nppiDiv\\_32f\\_C1IR, 291](#)  
[nppiDiv\\_32f\\_C1R, 291](#)  
[nppiDiv\\_32f\\_C3IR, 291](#)  
[nppiDiv\\_32f\\_C3R, 292](#)  
[nppiDiv\\_32f\\_C4IR, 292](#)  
[nppiDiv\\_32f\\_C4R, 292](#)  
[nppiDiv\\_32fc\\_AC4IR, 293](#)  
[nppiDiv\\_32fc\\_AC4R, 293](#)  
[nppiDiv\\_32fc\\_C1IR, 294](#)  
[nppiDiv\\_32fc\\_C1R, 294](#)  
[nppiDiv\\_32fc\\_C3IR, 294](#)  
[nppiDiv\\_32fc\\_C3R, 295](#)  
[nppiDiv\\_32fc\\_C4IR, 295](#)  
[nppiDiv\\_32fc\\_C4R, 295](#)  
[nppiDiv\\_32s\\_C1IRSfs, 296](#)  
[nppiDiv\\_32s\\_C1R, 296](#)  
[nppiDiv\\_32s\\_C1RSfs, 297](#)  
[nppiDiv\\_32s\\_C3IRSfs, 297](#)  
[nppiDiv\\_32s\\_C3RSfs, 297](#)  
[nppiDiv\\_32sc\\_AC4IRSfs, 298](#)  
[nppiDiv\\_32sc\\_AC4RSfs, 298](#)  
[nppiDiv\\_32sc\\_C1IRSfs, 299](#)  
[nppiDiv\\_32sc\\_C1RSfs, 299](#)  
[nppiDiv\\_32sc\\_C3IRSfs, 300](#)  
[nppiDiv\\_32sc\\_C3RSfs, 300](#)  
[nppiDiv\\_8u\\_AC4IRSfs, 300](#)  
[nppiDiv\\_8u\\_AC4RSfs, 301](#)  
[nppiDiv\\_8u\\_C1IRSfs, 301](#)  
[nppiDiv\\_8u\\_C1RSfs, 302](#)  
[nppiDiv\\_8u\\_C3IRSfs, 302](#)  
[nppiDiv\\_8u\\_C3RSfs, 302](#)  
[nppiDiv\\_8u\\_C4IRSfs, 303](#)  
[nppiDiv\\_8u\\_C4RSfs, 303](#)  
[image\\_divc](#)  
[nppiDivC\\_16s\\_AC4IRSfs, 145](#)  
[nppiDivC\\_16s\\_AC4RSfs, 145](#)  
[nppiDivC\\_16s\\_C1IRSfs, 145](#)  
[nppiDivC\\_16s\\_C1RSfs, 146](#)  
[nppiDivC\\_16s\\_C3IRSfs, 146](#)  
[nppiDivC\\_16s\\_C3RSfs, 146](#)  
[nppiDivC\\_16s\\_C4IRSfs, 147](#)  
[nppiDivC\\_16s\\_C4RSfs, 147](#)  
[nppiDivC\\_16sc\\_AC4IRSfs, 148](#)  
[nppiDivC\\_16sc\\_AC4RSfs, 148](#)  
[nppiDivC\\_16sc\\_C1IRSfs, 148](#)  
[nppiDivC\\_16sc\\_C1RSfs, 149](#)

- nppiDivC\_16sc\_C3IRSfs, [149](#)
- nppiDivC\_16sc\_C3RSfs, [150](#)
- nppiDivC\_16u\_AC4IRSfs, [150](#)
- nppiDivC\_16u\_AC4RSfs, [150](#)
- nppiDivC\_16u\_C1IRSfs, [151](#)
- nppiDivC\_16u\_C1RSfs, [151](#)
- nppiDivC\_16u\_C3IRSfs, [152](#)
- nppiDivC\_16u\_C3RSfs, [152](#)
- nppiDivC\_16u\_C4IRSfs, [152](#)
- nppiDivC\_16u\_C4RSfs, [153](#)
- nppiDivC\_32f\_AC4IR, [153](#)
- nppiDivC\_32f\_AC4R, [153](#)
- nppiDivC\_32f\_C1IR, [154](#)
- nppiDivC\_32f\_C1R, [154](#)
- nppiDivC\_32f\_C3IR, [154](#)
- nppiDivC\_32f\_C3R, [155](#)
- nppiDivC\_32f\_C4IR, [155](#)
- nppiDivC\_32f\_C4R, [155](#)
- nppiDivC\_32fc\_AC4IR, [156](#)
- nppiDivC\_32fc\_AC4R, [156](#)
- nppiDivC\_32fc\_C1IR, [156](#)
- nppiDivC\_32fc\_C1R, [157](#)
- nppiDivC\_32fc\_C3IR, [157](#)
- nppiDivC\_32fc\_C3R, [157](#)
- nppiDivC\_32fc\_C4IR, [158](#)
- nppiDivC\_32fc\_C4R, [158](#)
- nppiDivC\_32s\_C1IRSfs, [159](#)
- nppiDivC\_32s\_C1RSfs, [159](#)
- nppiDivC\_32s\_C3IRSfs, [159](#)
- nppiDivC\_32s\_C3RSfs, [160](#)
- nppiDivC\_32sc\_AC4IRSfs, [160](#)
- nppiDivC\_32sc\_AC4RSfs, [160](#)
- nppiDivC\_32sc\_C1IRSfs, [161](#)
- nppiDivC\_32sc\_C1RSfs, [161](#)
- nppiDivC\_32sc\_C3IRSfs, [162](#)
- nppiDivC\_32sc\_C3RSfs, [162](#)
- nppiDivC\_8u\_AC4IRSfs, [162](#)
- nppiDivC\_8u\_AC4RSfs, [163](#)
- nppiDivC\_8u\_C1IRSfs, [163](#)
- nppiDivC\_8u\_C1RSfs, [164](#)
- nppiDivC\_8u\_C3IRSfs, [164](#)
- nppiDivC\_8u\_C3RSfs, [164](#)
- nppiDivC\_8u\_C4IRSfs, [165](#)
- nppiDivC\_8u\_C4RSfs, [165](#)
- image\_divround
  - nppiDiv\_Round\_16s\_AC4IRSfs, [307](#)
  - nppiDiv\_Round\_16s\_AC4RSfs, [308](#)
  - nppiDiv\_Round\_16s\_C1IRSfs, [308](#)
  - nppiDiv\_Round\_16s\_C1RSfs, [309](#)
  - nppiDiv\_Round\_16s\_C3IRSfs, [309](#)
  - nppiDiv\_Round\_16s\_C3RSfs, [309](#)
  - nppiDiv\_Round\_16s\_C4IRSfs, [310](#)
  - nppiDiv\_Round\_16s\_C4RSfs, [310](#)
  - nppiDiv\_Round\_16u\_AC4IRSfs, [311](#)
  - nppiDiv\_Round\_16u\_AC4RSfs, [311](#)
  - nppiDiv\_Round\_16u\_C1IRSfs, [312](#)
  - nppiDiv\_Round\_16u\_C1RSfs, [312](#)
  - nppiDiv\_Round\_16u\_C3IRSfs, [313](#)
  - nppiDiv\_Round\_16u\_C3RSfs, [313](#)
  - nppiDiv\_Round\_16u\_C4IRSfs, [314](#)
  - nppiDiv\_Round\_16u\_C4RSfs, [314](#)
  - nppiDiv\_Round\_8u\_AC4IRSfs, [315](#)
  - nppiDiv\_Round\_8u\_AC4RSfs, [315](#)
  - nppiDiv\_Round\_8u\_C1IRSfs, [316](#)
  - nppiDiv\_Round\_8u\_C1RSfs, [316](#)
  - nppiDiv\_Round\_8u\_C3IRSfs, [317](#)
  - nppiDiv\_Round\_8u\_C3RSfs, [317](#)
  - nppiDiv\_Round\_8u\_C4IRSfs, [318](#)
  - nppiDiv\_Round\_8u\_C4RSfs, [318](#)
- image\_dot\_prod
  - nppiDotProd\_16s64f\_AC4R, [1805](#)
  - nppiDotProd\_16s64f\_C1R, [1805](#)
  - nppiDotProd\_16s64f\_C3R, [1806](#)
  - nppiDotProd\_16s64f\_C4R, [1806](#)
  - nppiDotProd\_16u64f\_AC4R, [1807](#)
  - nppiDotProd\_16u64f\_C1R, [1807](#)
  - nppiDotProd\_16u64f\_C3R, [1808](#)
  - nppiDotProd\_16u64f\_C4R, [1808](#)
  - nppiDotProd\_32f64f\_AC4R, [1808](#)
  - nppiDotProd\_32f64f\_C1R, [1809](#)
  - nppiDotProd\_32f64f\_C3R, [1809](#)
  - nppiDotProd\_32f64f\_C4R, [1810](#)
  - nppiDotProd\_32s64f\_AC4R, [1810](#)
  - nppiDotProd\_32s64f\_C1R, [1811](#)
  - nppiDotProd\_32s64f\_C3R, [1811](#)
  - nppiDotProd\_32s64f\_C4R, [1811](#)
  - nppiDotProd\_32u64f\_AC4R, [1812](#)
  - nppiDotProd\_32u64f\_C1R, [1812](#)
  - nppiDotProd\_32u64f\_C3R, [1813](#)
  - nppiDotProd\_32u64f\_C4R, [1813](#)
  - nppiDotProd\_8s64f\_AC4R, [1814](#)
  - nppiDotProd\_8s64f\_C1R, [1814](#)
  - nppiDotProd\_8s64f\_C3R, [1814](#)
  - nppiDotProd\_8s64f\_C4R, [1815](#)
  - nppiDotProd\_8u64f\_AC4R, [1815](#)
  - nppiDotProd\_8u64f\_C1R, [1816](#)
  - nppiDotProd\_8u64f\_C3R, [1816](#)
  - nppiDotProd\_8u64f\_C4R, [1817](#)
  - nppiDotProdGetBufferSize\_16s64f\_-  
AC4R, [1817](#)
  - nppiDotProdGetBufferSize\_16s64f\_C1R,  
[1817](#)
  - nppiDotProdGetBufferSize\_16s64f\_C3R,  
[1818](#)
  - nppiDotProdGetBufferSize\_16s64f\_C4R,  
[1818](#)
  - nppiDotProdGetBufferSize\_16u64f\_-  
AC4R, [1818](#)

- nppiDotProdGetBufferHostSize\_16u64f\_C1R, 1818
- nppiDotProdGetBufferHostSize\_16u64f\_C3R, 1819
- nppiDotProdGetBufferHostSize\_16u64f\_C4R, 1819
- nppiDotProdGetBufferHostSize\_32f64f\_-AC4R, 1819
- nppiDotProdGetBufferHostSize\_32f64f\_C1R, 1820
- nppiDotProdGetBufferHostSize\_32f64f\_C3R, 1820
- nppiDotProdGetBufferHostSize\_32f64f\_C4R, 1820
- nppiDotProdGetBufferHostSize\_32s64f\_-AC4R, 1820
- nppiDotProdGetBufferHostSize\_32s64f\_C1R, 1821
- nppiDotProdGetBufferHostSize\_32s64f\_C3R, 1821
- nppiDotProdGetBufferHostSize\_32s64f\_C4R, 1821
- nppiDotProdGetBufferHostSize\_32u64f\_-AC4R, 1822
- nppiDotProdGetBufferHostSize\_32u64f\_C1R, 1822
- nppiDotProdGetBufferHostSize\_32u64f\_C3R, 1822
- nppiDotProdGetBufferHostSize\_32u64f\_C4R, 1822
- nppiDotProdGetBufferHostSize\_8s64f\_-AC4R, 1823
- nppiDotProdGetBufferHostSize\_8s64f\_C1R, 1823
- nppiDotProdGetBufferHostSize\_8s64f\_C3R, 1823
- nppiDotProdGetBufferHostSize\_8s64f\_C4R, 1824
- nppiDotProdGetBufferHostSize\_8u64f\_-AC4R, 1824
- nppiDotProdGetBufferHostSize\_8u64f\_C1R, 1824
- nppiDotProdGetBufferHostSize\_8u64f\_C3R, 1824
- nppiDotProdGetBufferHostSize\_8u64f\_C4R, 1825
- image\_duplicate\_channel
  - nppiDup\_16s\_C1AC4R, 910
  - nppiDup\_16s\_C1C3R, 910
  - nppiDup\_16s\_C1C4R, 911
  - nppiDup\_16u\_C1AC4R, 911
  - nppiDup\_16u\_C1C3R, 911
  - nppiDup\_16u\_C1C4R, 912
  - nppiDup\_32f\_C1AC4R, 912
  - nppiDup\_32f\_C1C3R, 912
  - nppiDup\_32f\_C1C4R, 913
  - nppiDup\_32s\_C1AC4R, 913
  - nppiDup\_32s\_C1C3R, 913
  - nppiDup\_32s\_C1C4R, 914
  - nppiDup\_8u\_C1AC4R, 914
  - nppiDup\_8u\_C1C3R, 914
  - nppiDup\_8u\_C1C4R, 915
- image\_erode
  - nppiErode\_16u\_AC4R, 1375
  - nppiErode\_16u\_C1R, 1375
  - nppiErode\_16u\_C3R, 1376
  - nppiErode\_16u\_C4R, 1376
  - nppiErode\_32f\_AC4R, 1376
  - nppiErode\_32f\_C1R, 1377
  - nppiErode\_32f\_C3R, 1377
  - nppiErode\_32f\_C4R, 1378
  - nppiErode\_8u\_AC4R, 1378
  - nppiErode\_8u\_C1R, 1379
  - nppiErode\_8u\_C3R, 1379
  - nppiErode\_8u\_C4R, 1379
- image\_erode\_3x3
  - nppiErode3x3\_16u\_AC4R, 1388
  - nppiErode3x3\_16u\_C1R, 1388
  - nppiErode3x3\_16u\_C3R, 1388
  - nppiErode3x3\_16u\_C4R, 1389
  - nppiErode3x3\_32f\_AC4R, 1389
  - nppiErode3x3\_32f\_C1R, 1389
  - nppiErode3x3\_32f\_C3R, 1390
  - nppiErode3x3\_32f\_C4R, 1390
  - nppiErode3x3\_64f\_C1R, 1390
  - nppiErode3x3\_8u\_AC4R, 1391
  - nppiErode3x3\_8u\_C1R, 1391
  - nppiErode3x3\_8u\_C3R, 1391
  - nppiErode3x3\_8u\_C4R, 1392
- image\_exp
  - nppiExp\_16s\_C1IRSfs, 364
  - nppiExp\_16s\_C1RSfs, 364
  - nppiExp\_16s\_C3IRSfs, 365
  - nppiExp\_16s\_C3RSfs, 365
  - nppiExp\_16u\_C1IRSfs, 365
  - nppiExp\_16u\_C1RSfs, 366
  - nppiExp\_16u\_C3IRSfs, 366
  - nppiExp\_16u\_C3RSfs, 366
  - nppiExp\_32f\_C1IR, 367
  - nppiExp\_32f\_C1R, 367
  - nppiExp\_32f\_C3IR, 367
  - nppiExp\_32f\_C3R, 368
  - nppiExp\_8u\_C1IRSfs, 368
  - nppiExp\_8u\_C1RSfs, 368
  - nppiExp\_8u\_C3IRSfs, 369
  - nppiExp\_8u\_C3RSfs, 369
- image\_filtering\_functions
  - nppiFilterGauss\_16s\_AC4R, 951

- [nppiFilterGauss\\_16s\\_C1R](#), 952
- [nppiFilterGauss\\_16s\\_C3R](#), 952
- [nppiFilterGauss\\_16s\\_C4R](#), 952
- [nppiFilterGauss\\_16u\\_AC4R](#), 953
- [nppiFilterGauss\\_16u\\_C1R](#), 953
- [nppiFilterGauss\\_16u\\_C3R](#), 953
- [nppiFilterGauss\\_16u\\_C4R](#), 954
- [nppiFilterGauss\\_32f\\_AC4R](#), 954
- [nppiFilterGauss\\_32f\\_C1R](#), 954
- [nppiFilterGauss\\_32f\\_C3R](#), 955
- [nppiFilterGauss\\_32f\\_C4R](#), 955
- [nppiFilterGauss\\_8u\\_AC4R](#), 955
- [nppiFilterGauss\\_8u\\_C1R](#), 956
- [nppiFilterGauss\\_8u\\_C3R](#), 956
- [nppiFilterGauss\\_8u\\_C4R](#), 956
- [nppiFilterHighPass\\_16s\\_AC4R](#), 957
- [nppiFilterHighPass\\_16s\\_C1R](#), 957
- [nppiFilterHighPass\\_16s\\_C3R](#), 957
- [nppiFilterHighPass\\_16s\\_C4R](#), 958
- [nppiFilterHighPass\\_16u\\_AC4R](#), 958
- [nppiFilterHighPass\\_16u\\_C1R](#), 958
- [nppiFilterHighPass\\_16u\\_C3R](#), 959
- [nppiFilterHighPass\\_16u\\_C4R](#), 959
- [nppiFilterHighPass\\_32f\\_AC4R](#), 959
- [nppiFilterHighPass\\_32f\\_C1R](#), 960
- [nppiFilterHighPass\\_32f\\_C3R](#), 960
- [nppiFilterHighPass\\_32f\\_C4R](#), 960
- [nppiFilterHighPass\\_8u\\_AC4R](#), 961
- [nppiFilterHighPass\\_8u\\_C1R](#), 961
- [nppiFilterHighPass\\_8u\\_C3R](#), 961
- [nppiFilterHighPass\\_8u\\_C4R](#), 962
- [nppiFilterLaplace\\_16s\\_AC4R](#), 962
- [nppiFilterLaplace\\_16s\\_C1R](#), 962
- [nppiFilterLaplace\\_16s\\_C3R](#), 963
- [nppiFilterLaplace\\_16s\\_C4R](#), 963
- [nppiFilterLaplace\\_32f\\_AC4R](#), 963
- [nppiFilterLaplace\\_32f\\_C1R](#), 964
- [nppiFilterLaplace\\_32f\\_C3R](#), 964
- [nppiFilterLaplace\\_32f\\_C4R](#), 964
- [nppiFilterLaplace\\_8s16s\\_C1R](#), 965
- [nppiFilterLaplace\\_8u16s\\_C1R](#), 965
- [nppiFilterLaplace\\_8u\\_AC4R](#), 965
- [nppiFilterLaplace\\_8u\\_C1R](#), 966
- [nppiFilterLaplace\\_8u\\_C3R](#), 966
- [nppiFilterLaplace\\_8u\\_C4R](#), 966
- [nppiFilterLowPass\\_16s\\_AC4R](#), 967
- [nppiFilterLowPass\\_16s\\_C1R](#), 967
- [nppiFilterLowPass\\_16s\\_C3R](#), 967
- [nppiFilterLowPass\\_16s\\_C4R](#), 968
- [nppiFilterLowPass\\_16u\\_AC4R](#), 968
- [nppiFilterLowPass\\_16u\\_C1R](#), 968
- [nppiFilterLowPass\\_16u\\_C3R](#), 969
- [nppiFilterLowPass\\_16u\\_C4R](#), 969
- [nppiFilterLowPass\\_32f\\_AC4R](#), 969
- [nppiFilterLowPass\\_32f\\_C1R](#), 970
- [nppiFilterLowPass\\_32f\\_C3R](#), 970
- [nppiFilterLowPass\\_32f\\_C4R](#), 970
- [nppiFilterLowPass\\_8u\\_AC4R](#), 971
- [nppiFilterLowPass\\_8u\\_C1R](#), 971
- [nppiFilterLowPass\\_8u\\_C3R](#), 971
- [nppiFilterLowPass\\_8u\\_C4R](#), 972
- [nppiFilterRobertsDown\\_16s\\_AC4R](#), 972
- [nppiFilterRobertsDown\\_16s\\_C1R](#), 972
- [nppiFilterRobertsDown\\_16s\\_C3R](#), 973
- [nppiFilterRobertsDown\\_16s\\_C4R](#), 973
- [nppiFilterRobertsDown\\_32f\\_AC4R](#), 973
- [nppiFilterRobertsDown\\_32f\\_C1R](#), 974
- [nppiFilterRobertsDown\\_32f\\_C3R](#), 974
- [nppiFilterRobertsDown\\_32f\\_C4R](#), 974
- [nppiFilterRobertsDown\\_8u\\_AC4R](#), 975
- [nppiFilterRobertsDown\\_8u\\_C1R](#), 975
- [nppiFilterRobertsDown\\_8u\\_C3R](#), 975
- [nppiFilterRobertsDown\\_8u\\_C4R](#), 976
- [nppiFilterRobertsUp\\_16s\\_AC4R](#), 976
- [nppiFilterRobertsUp\\_16s\\_C1R](#), 976
- [nppiFilterRobertsUp\\_16s\\_C3R](#), 977
- [nppiFilterRobertsUp\\_16s\\_C4R](#), 977
- [nppiFilterRobertsUp\\_32f\\_AC4R](#), 977
- [nppiFilterRobertsUp\\_32f\\_C1R](#), 978
- [nppiFilterRobertsUp\\_32f\\_C3R](#), 978
- [nppiFilterRobertsUp\\_32f\\_C4R](#), 978
- [nppiFilterRobertsUp\\_8u\\_AC4R](#), 979
- [nppiFilterRobertsUp\\_8u\\_C1R](#), 979
- [nppiFilterRobertsUp\\_8u\\_C3R](#), 979
- [nppiFilterRobertsUp\\_8u\\_C4R](#), 980
- [nppiFilterSharpen\\_16s\\_AC4R](#), 980
- [nppiFilterSharpen\\_16s\\_C1R](#), 980
- [nppiFilterSharpen\\_16s\\_C3R](#), 981
- [nppiFilterSharpen\\_16s\\_C4R](#), 981
- [nppiFilterSharpen\\_16u\\_AC4R](#), 981
- [nppiFilterSharpen\\_16u\\_C1R](#), 982
- [nppiFilterSharpen\\_16u\\_C3R](#), 982
- [nppiFilterSharpen\\_16u\\_C4R](#), 982
- [nppiFilterSharpen\\_32f\\_AC4R](#), 983
- [nppiFilterSharpen\\_32f\\_C1R](#), 983
- [nppiFilterSharpen\\_32f\\_C3R](#), 983
- [nppiFilterSharpen\\_32f\\_C4R](#), 984
- [nppiFilterSharpen\\_8u\\_AC4R](#), 984
- [nppiFilterSharpen\\_8u\\_C1R](#), 984
- [nppiFilterSharpen\\_8u\\_C3R](#), 985
- [nppiFilterSharpen\\_8u\\_C4R](#), 985
- [nppiFilterSobelCrossBorder\\_32f\\_C1R](#), 985
- [nppiFilterSobelCrossBorder\\_8s16s\\_C1R](#), 986
- [nppiFilterSobelCrossBorder\\_8u16s\\_C1R](#), 986
- [nppiFilterSobelVertSecondBorder\\_32f\\_C1R](#), 987
- [nppiFilterSobelVertSecondBorder\\_8s16s\\_C1R](#), 987

[ippiFilterSobelVertSecondBorder\\_8u16s\\_C1R, 988](#)  
[image\\_fourier\\_transforms](#)  
     [ippiMagnitude\\_32fc32f\\_C1R, 1364](#)  
     [ippiMagnitudeSqr\\_32fc32f\\_C1R, 1364](#)  
[image\\_graphcut](#)  
     [ippiGraphcut8\\_32f8u, 713](#)  
     [ippiGraphcut8\\_32s8u, 713](#)  
     [ippiGraphcut8GetSize, 714](#)  
     [ippiGraphcut8InitAlloc, 715](#)  
     [ippiGraphcut\\_32f8u, 715](#)  
     [ippiGraphcut\\_32s8u, 716](#)  
     [ippiGraphcutFree, 717](#)  
     [ippiGraphcutGetSize, 717](#)  
     [ippiGraphcutInitAlloc, 717](#)  
[image\\_histogrameven](#)  
     [ippiEvenLevelsHost\\_32s, 1856](#)  
     [ippiHistogramEven\\_16s\\_AC4R, 1857](#)  
     [ippiHistogramEven\\_16s\\_C1R, 1857](#)  
     [ippiHistogramEven\\_16s\\_C3R, 1857](#)  
     [ippiHistogramEven\\_16s\\_C4R, 1858](#)  
     [ippiHistogramEven\\_16u\\_AC4R, 1858](#)  
     [ippiHistogramEven\\_16u\\_C1R, 1859](#)  
     [ippiHistogramEven\\_16u\\_C3R, 1859](#)  
     [ippiHistogramEven\\_16u\\_C4R, 1860](#)  
     [ippiHistogramEven\\_8u\\_AC4R, 1860](#)  
     [ippiHistogramEven\\_8u\\_C1R, 1861](#)  
     [ippiHistogramEven\\_8u\\_C3R, 1861](#)  
     [ippiHistogramEven\\_8u\\_C4R, 1862](#)  
     [ippiHistogramEvenGetBufferSize\\_16s\\_AC4R, 1862](#)  
     [ippiHistogramEvenGetBufferSize\\_16s\\_C1R, 1862](#)  
     [ippiHistogramEvenGetBufferSize\\_16s\\_C3R, 1863](#)  
     [ippiHistogramEvenGetBufferSize\\_16s\\_C4R, 1863](#)  
     [ippiHistogramEvenGetBufferSize\\_16u\\_AC4R, 1863](#)  
     [ippiHistogramEvenGetBufferSize\\_16u\\_C1R, 1864](#)  
     [ippiHistogramEvenGetBufferSize\\_16u\\_C3R, 1864](#)  
     [ippiHistogramEvenGetBufferSize\\_16u\\_C4R, 1864](#)  
     [ippiHistogramEvenGetBufferSize\\_8u\\_AC4R, 1865](#)  
     [ippiHistogramEvenGetBufferSize\\_8u\\_C1R, 1865](#)  
     [ippiHistogramEvenGetBufferSize\\_8u\\_C3R, 1865](#)  
     [ippiHistogramEvenGetBufferSize\\_8u\\_C4R, 1866](#)  
[image\\_histogramrange](#)



- nppiNorm\_Inf\_16u\_C1MR, [1604](#)
- nppiNorm\_Inf\_16u\_C1R, [1605](#)
- nppiNorm\_Inf\_16u\_C3CMR, [1605](#)
- nppiNorm\_Inf\_16u\_C3R, [1606](#)
- nppiNorm\_Inf\_16u\_C4R, [1606](#)
- nppiNorm\_Inf\_32f\_AC4R, [1606](#)
- nppiNorm\_Inf\_32f\_C1MR, [1607](#)
- nppiNorm\_Inf\_32f\_C1R, [1607](#)
- nppiNorm\_Inf\_32f\_C3CMR, [1608](#)
- nppiNorm\_Inf\_32f\_C3R, [1608](#)
- nppiNorm\_Inf\_32f\_C4R, [1608](#)
- nppiNorm\_Inf\_32s\_C1R, [1609](#)
- nppiNorm\_Inf\_8s\_C1MR, [1609](#)
- nppiNorm\_Inf\_8s\_C3CMR, [1610](#)
- nppiNorm\_Inf\_8u\_AC4R, [1610](#)
- nppiNorm\_Inf\_8u\_C1MR, [1610](#)
- nppiNorm\_Inf\_8u\_C1R, [1611](#)
- nppiNorm\_Inf\_8u\_C3CMR, [1611](#)
- nppiNorm\_Inf\_8u\_C3R, [1612](#)
- nppiNorm\_Inf\_8u\_C4R, [1612](#)
- nppiNormInfGetBufferHostSize\_16s\_AC4R, [1612](#)
- nppiNormInfGetBufferHostSize\_16s\_C1R, [1613](#)
- nppiNormInfGetBufferHostSize\_16s\_C3R, [1613](#)
- nppiNormInfGetBufferHostSize\_16s\_C4R, [1613](#)
- nppiNormInfGetBufferHostSize\_16u\_AC4R, [1614](#)
- nppiNormInfGetBufferHostSize\_16u\_C1MR, [1614](#)
- nppiNormInfGetBufferHostSize\_16u\_C1R, [1614](#)
- nppiNormInfGetBufferHostSize\_16u\_-C3CMR, [1614](#)
- nppiNormInfGetBufferHostSize\_16u\_C3R, [1615](#)
- nppiNormInfGetBufferHostSize\_16u\_C4R, [1615](#)
- nppiNormInfGetBufferHostSize\_32f\_AC4R, [1615](#)
- nppiNormInfGetBufferHostSize\_32f\_C1MR, [1616](#)
- nppiNormInfGetBufferHostSize\_32f\_C1R, [1616](#)
- nppiNormInfGetBufferHostSize\_32f\_-C3CMR, [1616](#)
- nppiNormInfGetBufferHostSize\_32f\_C3R, [1616](#)
- nppiNormInfGetBufferHostSize\_32f\_C4R, [1617](#)
- nppiNormInfGetBufferHostSize\_32s\_C1R, [1617](#)
- nppiNormInfGetBufferHostSize\_8s\_C1MR, [1617](#)
- nppiNormInfGetBufferHostSize\_8s\_C3CMR, [1618](#)
- nppiNormInfGetBufferHostSize\_8u\_AC4R, [1618](#)
- nppiNormInfGetBufferHostSize\_8u\_C1MR, [1618](#)
- nppiNormInfGetBufferHostSize\_8u\_C1R, [1618](#)
- nppiNormInfGetBufferHostSize\_8u\_C3CMR, [1619](#)
- nppiNormInfGetBufferHostSize\_8u\_C3R, [1619](#)
- nppiNormInfGetBufferHostSize\_8u\_C4R, [1619](#)
- image\_inf\_normdiff
  - nppiNormDiff\_Inf\_16s\_AC4R, [1667](#)
  - nppiNormDiff\_Inf\_16s\_C1R, [1667](#)
  - nppiNormDiff\_Inf\_16s\_C3R, [1668](#)
  - nppiNormDiff\_Inf\_16s\_C4R, [1668](#)
  - nppiNormDiff\_Inf\_16u\_AC4R, [1669](#)
  - nppiNormDiff\_Inf\_16u\_C1MR, [1669](#)
  - nppiNormDiff\_Inf\_16u\_C1R, [1670](#)
  - nppiNormDiff\_Inf\_16u\_C3CMR, [1670](#)
  - nppiNormDiff\_Inf\_16u\_C3R, [1671](#)
  - nppiNormDiff\_Inf\_16u\_C4R, [1671](#)
  - nppiNormDiff\_Inf\_32f\_AC4R, [1671](#)
  - nppiNormDiff\_Inf\_32f\_C1MR, [1672](#)
  - nppiNormDiff\_Inf\_32f\_C1R, [1672](#)
  - nppiNormDiff\_Inf\_32f\_C3CMR, [1673](#)
  - nppiNormDiff\_Inf\_32f\_C3R, [1673](#)
  - nppiNormDiff\_Inf\_32f\_C4R, [1674](#)
  - nppiNormDiff\_Inf\_8s\_C1MR, [1674](#)
  - nppiNormDiff\_Inf\_8s\_C3CMR, [1675](#)
  - nppiNormDiff\_Inf\_8u\_AC4R, [1675](#)
  - nppiNormDiff\_Inf\_8u\_C1MR, [1676](#)
  - nppiNormDiff\_Inf\_8u\_C1R, [1676](#)
  - nppiNormDiff\_Inf\_8u\_C3CMR, [1677](#)
  - nppiNormDiff\_Inf\_8u\_C3R, [1677](#)
  - nppiNormDiff\_Inf\_8u\_C4R, [1678](#)
  - nppiNormDiffInfGetBufferHostSize\_16s\_-AC4R, [1678](#)
  - nppiNormDiffInfGetBufferHostSize\_16s\_-C1R, [1678](#)
  - nppiNormDiffInfGetBufferHostSize\_16s\_-C3R, [1679](#)
  - nppiNormDiffInfGetBufferHostSize\_16s\_-C4R, [1679](#)
  - nppiNormDiffInfGetBufferHostSize\_16u\_-AC4R, [1679](#)
  - nppiNormDiffInfGetBufferHostSize\_16u\_-C1MR, [1680](#)

- nppiNormDiffInfGetBufferHostSize\_16u\_-C1R, [1680](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_-C3CMR, [1680](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_-C3R, [1680](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_-C4R, [1681](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_-AC4R, [1681](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_-C1MR, [1681](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_-C1R, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_-C3CMR, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_-C3R, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_-C4R, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_8s\_-C1MR, [1683](#)
- nppiNormDiffInfGetBufferHostSize\_8s\_-C3CMR, [1683](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_-AC4R, [1683](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_-C1MR, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C1R, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_-C3CMR, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C3R, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C4R, [1685](#)
- image\_inf\_normrel
  - nppiNormRel\_Inf\_16s\_AC4R, [1736](#)
  - nppiNormRel\_Inf\_16s\_C1R, [1736](#)
  - nppiNormRel\_Inf\_16s\_C3R, [1737](#)
  - nppiNormRel\_Inf\_16s\_C4R, [1737](#)
  - nppiNormRel\_Inf\_16u\_AC4R, [1738](#)
  - nppiNormRel\_Inf\_16u\_C1MR, [1738](#)
  - nppiNormRel\_Inf\_16u\_C1R, [1739](#)
  - nppiNormRel\_Inf\_16u\_C3CMR, [1739](#)
  - nppiNormRel\_Inf\_16u\_C3R, [1740](#)
  - nppiNormRel\_Inf\_16u\_C4R, [1740](#)
  - nppiNormRel\_Inf\_32f\_AC4R, [1740](#)
  - nppiNormRel\_Inf\_32f\_C1MR, [1741](#)
  - nppiNormRel\_Inf\_32f\_C1R, [1741](#)
  - nppiNormRel\_Inf\_32f\_C3CMR, [1742](#)
  - nppiNormRel\_Inf\_32f\_C3R, [1742](#)
  - nppiNormRel\_Inf\_32f\_C4R, [1743](#)
  - nppiNormRel\_Inf\_8s\_C1MR, [1743](#)
  - nppiNormRel\_Inf\_8s\_C3CMR, [1744](#)
  - nppiNormRel\_Inf\_8u\_AC4R, [1744](#)
  - nppiNormRel\_Inf\_8u\_C1MR, [1745](#)
  - nppiNormRel\_Inf\_8u\_C1R, [1745](#)
  - nppiNormRel\_Inf\_8u\_C3CMR, [1746](#)
  - nppiNormRel\_Inf\_8u\_C3R, [1746](#)
  - nppiNormRel\_Inf\_8u\_C4R, [1747](#)
  - nppiNormRelInfGetBufferHostSize\_16s\_-AC4R, [1747](#)
  - nppiNormRelInfGetBufferHostSize\_16s\_-C1R, [1748](#)
  - nppiNormRelInfGetBufferHostSize\_16s\_-C3R, [1748](#)
  - nppiNormRelInfGetBufferHostSize\_16s\_-C4R, [1748](#)
  - nppiNormRelInfGetBufferHostSize\_16u\_-AC4R, [1748](#)
  - nppiNormRelInfGetBufferHostSize\_16u\_-C1MR, [1749](#)
  - nppiNormRelInfGetBufferHostSize\_16u\_-C1R, [1749](#)
  - nppiNormRelInfGetBufferHostSize\_16u\_-C3CMR, [1749](#)
  - nppiNormRelInfGetBufferHostSize\_16u\_-C3R, [1750](#)
  - nppiNormRelInfGetBufferHostSize\_16u\_-C4R, [1750](#)
  - nppiNormRelInfGetBufferHostSize\_32f\_-AC4R, [1750](#)
  - nppiNormRelInfGetBufferHostSize\_32f\_-C1MR, [1750](#)
  - nppiNormRelInfGetBufferHostSize\_32f\_C1R, [1751](#)
  - nppiNormRelInfGetBufferHostSize\_32f\_-C3CMR, [1751](#)
  - nppiNormRelInfGetBufferHostSize\_32f\_C3R, [1751](#)
  - nppiNormRelInfGetBufferHostSize\_32f\_C4R, [1752](#)
  - nppiNormRelInfGetBufferHostSize\_32s\_-C1R, [1752](#)
  - nppiNormRelInfGetBufferHostSize\_8s\_-C1MR, [1752](#)
  - nppiNormRelInfGetBufferHostSize\_8s\_-C3CMR, [1752](#)
  - nppiNormRelInfGetBufferHostSize\_8u\_-AC4R, [1753](#)
  - nppiNormRelInfGetBufferHostSize\_8u\_-C1MR, [1753](#)
  - nppiNormRelInfGetBufferHostSize\_8u\_C1R, [1753](#)
  - nppiNormRelInfGetBufferHostSize\_8u\_-C3CMR, [1754](#)

- nppiNormRelInfGetBufferHostSize\_8u\_C3R, 1754
- nppiNormRelInfGetBufferHostSize\_8u\_C4R, 1754
- image\_integral
  - nppiIntegral\_8u32f\_C1R, 1846
  - nppiIntegral\_8u32s\_C1R, 1846
- image\_L1\_norm
  - nppiNorm\_L1\_16s\_AC4R, 1625
  - nppiNorm\_L1\_16s\_C1R, 1625
  - nppiNorm\_L1\_16s\_C3R, 1625
  - nppiNorm\_L1\_16s\_C4R, 1626
  - nppiNorm\_L1\_16u\_AC4R, 1626
  - nppiNorm\_L1\_16u\_C1MR, 1626
  - nppiNorm\_L1\_16u\_C1R, 1627
  - nppiNorm\_L1\_16u\_C3CMR, 1627
  - nppiNorm\_L1\_16u\_C3R, 1628
  - nppiNorm\_L1\_16u\_C4R, 1628
  - nppiNorm\_L1\_32f\_AC4R, 1628
  - nppiNorm\_L1\_32f\_C1MR, 1629
  - nppiNorm\_L1\_32f\_C1R, 1629
  - nppiNorm\_L1\_32f\_C3CMR, 1629
  - nppiNorm\_L1\_32f\_C3R, 1630
  - nppiNorm\_L1\_32f\_C4R, 1630
  - nppiNorm\_L1\_8s\_C1MR, 1631
  - nppiNorm\_L1\_8s\_C3CMR, 1631
  - nppiNorm\_L1\_8u\_AC4R, 1631
  - nppiNorm\_L1\_8u\_C1MR, 1632
  - nppiNorm\_L1\_8u\_C1R, 1632
  - nppiNorm\_L1\_8u\_C3CMR, 1633
  - nppiNorm\_L1\_8u\_C3R, 1633
  - nppiNorm\_L1\_8u\_C4R, 1633
  - nppiNormL1GetBufferHostSize\_16s\_AC4R, 1634
  - nppiNormL1GetBufferHostSize\_16s\_C1R, 1634
  - nppiNormL1GetBufferHostSize\_16s\_C3R, 1634
  - nppiNormL1GetBufferHostSize\_16s\_C4R, 1635
  - nppiNormL1GetBufferHostSize\_16u\_AC4R, 1635
  - nppiNormL1GetBufferHostSize\_16u\_C1MR, 1635
  - nppiNormL1GetBufferHostSize\_16u\_C1R, 1636
  - nppiNormL1GetBufferHostSize\_16u\_-C3CMR, 1636
  - nppiNormL1GetBufferHostSize\_16u\_C3R, 1636
  - nppiNormL1GetBufferHostSize\_16u\_C4R, 1636
  - nppiNormL1GetBufferHostSize\_32f\_AC4R, 1637
- nppiNormL1GetBufferHostSize\_32f\_C1MR, 1637
- nppiNormL1GetBufferHostSize\_32f\_C1R, 1637
- nppiNormL1GetBufferHostSize\_32f\_-C3CMR, 1638
- nppiNormL1GetBufferHostSize\_32f\_C3R, 1638
- nppiNormL1GetBufferHostSize\_32f\_C4R, 1638
- nppiNormL1GetBufferHostSize\_8s\_C1MR, 1638
- nppiNormL1GetBufferHostSize\_8s\_C3CMR, 1639
- nppiNormL1GetBufferHostSize\_8u\_AC4R, 1639
- nppiNormL1GetBufferHostSize\_8u\_C1MR, 1639
- nppiNormL1GetBufferHostSize\_8u\_C1R, 1640
- nppiNormL1GetBufferHostSize\_8u\_C3CMR, 1640
- nppiNormL1GetBufferHostSize\_8u\_C3R, 1640
- nppiNormL1GetBufferHostSize\_8u\_C4R, 1640
- image\_L1\_normdiff
  - nppiNormDiff\_L1\_16s\_AC4R, 1690
  - nppiNormDiff\_L1\_16s\_C1R, 1690
  - nppiNormDiff\_L1\_16s\_C3R, 1691
  - nppiNormDiff\_L1\_16s\_C4R, 1691
  - nppiNormDiff\_L1\_16u\_AC4R, 1692
  - nppiNormDiff\_L1\_16u\_C1MR, 1692
  - nppiNormDiff\_L1\_16u\_C1R, 1692
  - nppiNormDiff\_L1\_16u\_C3CMR, 1693
  - nppiNormDiff\_L1\_16u\_C3R, 1693
  - nppiNormDiff\_L1\_16u\_C4R, 1694
  - nppiNormDiff\_L1\_32f\_AC4R, 1694
  - nppiNormDiff\_L1\_32f\_C1MR, 1695
  - nppiNormDiff\_L1\_32f\_C1R, 1695
  - nppiNormDiff\_L1\_32f\_C3CMR, 1696
  - nppiNormDiff\_L1\_32f\_C3R, 1696
  - nppiNormDiff\_L1\_32f\_C4R, 1697
  - nppiNormDiff\_L1\_8s\_C1MR, 1697
  - nppiNormDiff\_L1\_8s\_C3CMR, 1698
  - nppiNormDiff\_L1\_8u\_AC4R, 1698
  - nppiNormDiff\_L1\_8u\_C1MR, 1699
  - nppiNormDiff\_L1\_8u\_C1R, 1699
  - nppiNormDiff\_L1\_8u\_C3CMR, 1699
  - nppiNormDiff\_L1\_8u\_C3R, 1700
  - nppiNormDiff\_L1\_8u\_C4R, 1700
  - nppiNormDiffL1GetBufferHostSize\_16s\_-AC4R, 1701



- nppiNormDiffL1GetBufferHostSize\_16s\_-C1R, [1701](#)
- nppiNormDiffL1GetBufferHostSize\_16s\_-C3R, [1701](#)
- nppiNormDiffL1GetBufferHostSize\_16s\_-C4R, [1702](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_-AC4R, [1702](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_-C1MR, [1702](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_-C1R, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_-C3CMR, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_-C3R, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_-C4R, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_-AC4R, [1704](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_-C1MR, [1704](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_-C1R, [1704](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_-C3CMR, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_-C3R, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_-C4R, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_8s\_-C1MR, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_8s\_-C3CMR, [1706](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_-AC4R, [1706](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_-C1MR, [1706](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C1R, [1707](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_-C3CMR, [1707](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C3R, [1707](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C4R, [1707](#)
- image\_L1\_normrel
  - nppiNormRel\_L1\_16s\_AC4R, [1759](#)
  - nppiNormRel\_L1\_16s\_C1R, [1759](#)
  - nppiNormRel\_L1\_16s\_C3R, [1760](#)
  - nppiNormRel\_L1\_16s\_C4R, [1760](#)
  - nppiNormRel\_L1\_16u\_AC4R, [1761](#)
  - nppiNormRel\_L1\_16u\_C1MR, [1761](#)
  - nppiNormRel\_L1\_16u\_C1R, [1762](#)
  - nppiNormRel\_L1\_16u\_C3CMR, [1762](#)
  - nppiNormRel\_L1\_16u\_C3R, [1762](#)
  - nppiNormRel\_L1\_16u\_C4R, [1763](#)
  - nppiNormRel\_L1\_32f\_AC4R, [1763](#)
  - nppiNormRel\_L1\_32f\_C1MR, [1764](#)
  - nppiNormRel\_L1\_32f\_C1R, [1764](#)
  - nppiNormRel\_L1\_32f\_C3CMR, [1765](#)
  - nppiNormRel\_L1\_32f\_C3R, [1765](#)
  - nppiNormRel\_L1\_32f\_C4R, [1766](#)
  - nppiNormRel\_L1\_8s\_C1MR, [1766](#)
  - nppiNormRel\_L1\_8s\_C3CMR, [1767](#)
  - nppiNormRel\_L1\_8u\_AC4R, [1767](#)
  - nppiNormRel\_L1\_8u\_C1MR, [1768](#)
  - nppiNormRel\_L1\_8u\_C1R, [1768](#)
  - nppiNormRel\_L1\_8u\_C3CMR, [1769](#)
  - nppiNormRel\_L1\_8u\_C3R, [1769](#)
  - nppiNormRel\_L1\_8u\_C4R, [1770](#)
  - nppiNormRelL1GetBufferHostSize\_16s\_-AC4R, [1770](#)
  - nppiNormRelL1GetBufferHostSize\_16s\_C1R, [1770](#)
  - nppiNormRelL1GetBufferHostSize\_16s\_C3R, [1771](#)
  - nppiNormRelL1GetBufferHostSize\_16s\_C4R, [1771](#)
  - nppiNormRelL1GetBufferHostSize\_16u\_-AC4R, [1771](#)
  - nppiNormRelL1GetBufferHostSize\_16u\_-C1MR, [1772](#)
  - nppiNormRelL1GetBufferHostSize\_16u\_-C1R, [1772](#)
  - nppiNormRelL1GetBufferHostSize\_16u\_-C3CMR, [1772](#)
  - nppiNormRelL1GetBufferHostSize\_16u\_-C3R, [1772](#)
  - nppiNormRelL1GetBufferHostSize\_16u\_-C4R, [1773](#)
  - nppiNormRelL1GetBufferHostSize\_32f\_-AC4R, [1773](#)
  - nppiNormRelL1GetBufferHostSize\_32f\_-C1MR, [1773](#)
  - nppiNormRelL1GetBufferHostSize\_32f\_C1R, [1774](#)
  - nppiNormRelL1GetBufferHostSize\_32f\_-C3CMR, [1774](#)
  - nppiNormRelL1GetBufferHostSize\_32f\_C3R, [1774](#)
  - nppiNormRelL1GetBufferHostSize\_32f\_C4R, [1774](#)
  - nppiNormRelL1GetBufferHostSize\_8s\_-C1MR, [1775](#)
  - nppiNormRelL1GetBufferHostSize\_8s\_-C3CMR, [1775](#)

- nppiNormRelL1GetBufferHostSize\_8u\_-AC4R, [1775](#)
- nppiNormRelL1GetBufferHostSize\_8u\_-C1MR, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C1R, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_-C3CMR, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C3R, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C4R, [1777](#)
- image\_L2\_norm
  - nppiNorm\_L2\_16s\_AC4R, [1646](#)
  - nppiNorm\_L2\_16s\_C1R, [1646](#)
  - nppiNorm\_L2\_16s\_C3R, [1646](#)
  - nppiNorm\_L2\_16s\_C4R, [1647](#)
  - nppiNorm\_L2\_16u\_AC4R, [1647](#)
  - nppiNorm\_L2\_16u\_C1MR, [1647](#)
  - nppiNorm\_L2\_16u\_C1R, [1648](#)
  - nppiNorm\_L2\_16u\_C3CMR, [1648](#)
  - nppiNorm\_L2\_16u\_C3R, [1649](#)
  - nppiNorm\_L2\_16u\_C4R, [1649](#)
  - nppiNorm\_L2\_32f\_AC4R, [1649](#)
  - nppiNorm\_L2\_32f\_C1MR, [1650](#)
  - nppiNorm\_L2\_32f\_C1R, [1650](#)
  - nppiNorm\_L2\_32f\_C3CMR, [1650](#)
  - nppiNorm\_L2\_32f\_C3R, [1651](#)
  - nppiNorm\_L2\_32f\_C4R, [1651](#)
  - nppiNorm\_L2\_8s\_C1MR, [1652](#)
  - nppiNorm\_L2\_8s\_C3CMR, [1652](#)
  - nppiNorm\_L2\_8u\_AC4R, [1652](#)
  - nppiNorm\_L2\_8u\_C1MR, [1653](#)
  - nppiNorm\_L2\_8u\_C1R, [1653](#)
  - nppiNorm\_L2\_8u\_C3CMR, [1654](#)
  - nppiNorm\_L2\_8u\_C3R, [1654](#)
  - nppiNorm\_L2\_8u\_C4R, [1654](#)
  - nppiNormL2GetBufferHostSize\_16s\_AC4R, [1655](#)
  - nppiNormL2GetBufferHostSize\_16s\_C1R, [1655](#)
  - nppiNormL2GetBufferHostSize\_16s\_C3R, [1655](#)
  - nppiNormL2GetBufferHostSize\_16s\_C4R, [1656](#)
  - nppiNormL2GetBufferHostSize\_16u\_AC4R, [1656](#)
  - nppiNormL2GetBufferHostSize\_16u\_C1MR, [1656](#)
  - nppiNormL2GetBufferHostSize\_16u\_C1R, [1657](#)
  - nppiNormL2GetBufferHostSize\_16u\_-C3CMR, [1657](#)
  - nppiNormL2GetBufferHostSize\_16u\_C3R, [1657](#)
  - nppiNormL2GetBufferHostSize\_16u\_C4R, [1657](#)
  - nppiNormL2GetBufferHostSize\_32f\_AC4R, [1658](#)
  - nppiNormL2GetBufferHostSize\_32f\_C1MR, [1658](#)
  - nppiNormL2GetBufferHostSize\_32f\_C1R, [1658](#)
  - nppiNormL2GetBufferHostSize\_32f\_-C3CMR, [1659](#)
  - nppiNormL2GetBufferHostSize\_32f\_C3R, [1659](#)
  - nppiNormL2GetBufferHostSize\_32f\_C4R, [1659](#)
  - nppiNormL2GetBufferHostSize\_8s\_C1MR, [1659](#)
  - nppiNormL2GetBufferHostSize\_8s\_C3CMR, [1660](#)
  - nppiNormL2GetBufferHostSize\_8u\_AC4R, [1660](#)
  - nppiNormL2GetBufferHostSize\_8u\_C1MR, [1660](#)
  - nppiNormL2GetBufferHostSize\_8u\_C1R, [1661](#)
  - nppiNormL2GetBufferHostSize\_8u\_C3CMR, [1661](#)
  - nppiNormL2GetBufferHostSize\_8u\_C3R, [1661](#)
  - nppiNormL2GetBufferHostSize\_8u\_C4R, [1661](#)
- image\_L2\_normdiff
  - nppiNormDiff\_L2\_16s\_AC4R, [1713](#)
  - nppiNormDiff\_L2\_16s\_C1R, [1713](#)
  - nppiNormDiff\_L2\_16s\_C3R, [1714](#)
  - nppiNormDiff\_L2\_16s\_C4R, [1714](#)
  - nppiNormDiff\_L2\_16u\_AC4R, [1715](#)
  - nppiNormDiff\_L2\_16u\_C1MR, [1715](#)
  - nppiNormDiff\_L2\_16u\_C1R, [1715](#)
  - nppiNormDiff\_L2\_16u\_C3CMR, [1716](#)
  - nppiNormDiff\_L2\_16u\_C3R, [1716](#)
  - nppiNormDiff\_L2\_16u\_C4R, [1717](#)
  - nppiNormDiff\_L2\_32f\_AC4R, [1717](#)
  - nppiNormDiff\_L2\_32f\_C1MR, [1718](#)
  - nppiNormDiff\_L2\_32f\_C1R, [1718](#)
  - nppiNormDiff\_L2\_32f\_C3CMR, [1719](#)
  - nppiNormDiff\_L2\_32f\_C3R, [1719](#)
  - nppiNormDiff\_L2\_32f\_C4R, [1720](#)
  - nppiNormDiff\_L2\_8s\_C1MR, [1720](#)
  - nppiNormDiff\_L2\_8s\_C3CMR, [1721](#)
  - nppiNormDiff\_L2\_8u\_AC4R, [1721](#)
  - nppiNormDiff\_L2\_8u\_C1MR, [1722](#)
  - nppiNormDiff\_L2\_8u\_C1R, [1722](#)

- nppiNormDiff\_L2\_8u\_C3CMR, [1722](#)
- nppiNormDiff\_L2\_8u\_C3R, [1723](#)
- nppiNormDiff\_L2\_8u\_C4R, [1723](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_-AC4R, [1724](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_-C1R, [1724](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_-C3R, [1724](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_-C4R, [1725](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_-AC4R, [1725](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_-C1MR, [1725](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_-C1R, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_-C3CMR, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_-C3R, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_-C4R, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_-AC4R, [1727](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_-C1MR, [1727](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_-C1R, [1727](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_-C3CMR, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_-C3R, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_-C4R, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_8s\_-C1MR, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_8s\_-C3CMR, [1729](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_-AC4R, [1729](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_-C1MR, [1729](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_C1R, [1730](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_-C3CMR, [1730](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_C3R, [1730](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_C4R, [1730](#)
- image\_L2\_normrel
  - nppiNormRel\_L2\_16s\_AC4R, [1782](#)
  - nppiNormRel\_L2\_16s\_C1R, [1782](#)
  - nppiNormRel\_L2\_16s\_C3R, [1783](#)
  - nppiNormRel\_L2\_16s\_C4R, [1783](#)
  - nppiNormRel\_L2\_16u\_AC4R, [1784](#)
  - nppiNormRel\_L2\_16u\_C1MR, [1784](#)
  - nppiNormRel\_L2\_16u\_C1R, [1785](#)
  - nppiNormRel\_L2\_16u\_C3CMR, [1785](#)
  - nppiNormRel\_L2\_16u\_C3R, [1785](#)
  - nppiNormRel\_L2\_16u\_C4R, [1786](#)
  - nppiNormRel\_L2\_32f\_AC4R, [1786](#)
  - nppiNormRel\_L2\_32f\_C1MR, [1787](#)
  - nppiNormRel\_L2\_32f\_C1R, [1787](#)
  - nppiNormRel\_L2\_32f\_C3CMR, [1788](#)
  - nppiNormRel\_L2\_32f\_C3R, [1788](#)
  - nppiNormRel\_L2\_32f\_C4R, [1789](#)
  - nppiNormRel\_L2\_8s\_C1MR, [1789](#)
  - nppiNormRel\_L2\_8s\_C3CMR, [1790](#)
  - nppiNormRel\_L2\_8u\_AC4R, [1790](#)
  - nppiNormRel\_L2\_8u\_C1MR, [1791](#)
  - nppiNormRel\_L2\_8u\_C1R, [1791](#)
  - nppiNormRel\_L2\_8u\_C3CMR, [1792](#)
  - nppiNormRel\_L2\_8u\_C3R, [1792](#)
  - nppiNormRel\_L2\_8u\_C4R, [1793](#)
  - nppiNormRelL2GetBufferHostSize\_16s\_-AC4R, [1793](#)
  - nppiNormRelL2GetBufferHostSize\_16s\_C1R, [1793](#)
  - nppiNormRelL2GetBufferHostSize\_16s\_C3R, [1794](#)
  - nppiNormRelL2GetBufferHostSize\_16s\_C4R, [1794](#)
  - nppiNormRelL2GetBufferHostSize\_16u\_-AC4R, [1794](#)
  - nppiNormRelL2GetBufferHostSize\_16u\_-C1MR, [1795](#)
  - nppiNormRelL2GetBufferHostSize\_16u\_-C1R, [1795](#)
  - nppiNormRelL2GetBufferHostSize\_16u\_-C3CMR, [1795](#)
  - nppiNormRelL2GetBufferHostSize\_16u\_-C3R, [1795](#)
  - nppiNormRelL2GetBufferHostSize\_16u\_-C4R, [1796](#)
  - nppiNormRelL2GetBufferHostSize\_32f\_-AC4R, [1796](#)
  - nppiNormRelL2GetBufferHostSize\_32f\_-C1MR, [1796](#)
  - nppiNormRelL2GetBufferHostSize\_32f\_C1R, [1797](#)
  - nppiNormRelL2GetBufferHostSize\_32f\_-C3CMR, [1797](#)
  - nppiNormRelL2GetBufferHostSize\_32f\_C3R, [1797](#)
  - nppiNormRelL2GetBufferHostSize\_32f\_C4R, [1797](#)

- nppiNormRelL2GetBufferHostSize\_8s\_-C1MR, 1798
- nppiNormRelL2GetBufferHostSize\_8s\_-C3CMR, 1798
- nppiNormRelL2GetBufferHostSize\_8u\_-AC4R, 1798
- nppiNormRelL2GetBufferHostSize\_8u\_-C1MR, 1799
- nppiNormRelL2GetBufferHostSize\_8u\_C1R, 1799
- nppiNormRelL2GetBufferHostSize\_8u\_-C3CMR, 1799
- nppiNormRelL2GetBufferHostSize\_8u\_C3R, 1799
- nppiNormRelL2GetBufferHostSize\_8u\_C4R, 1800
- image\_labeling\_and\_segmentation
  - NppiGraphcutState, 711
- image\_ln
  - nppiLn\_16s\_C1IRSfs, 357
  - nppiLn\_16s\_C1RSfs, 357
  - nppiLn\_16s\_C3IRSfs, 358
  - nppiLn\_16s\_C3RSfs, 358
  - nppiLn\_16u\_C1IRSfs, 358
  - nppiLn\_16u\_C1RSfs, 359
  - nppiLn\_16u\_C3IRSfs, 359
  - nppiLn\_16u\_C3RSfs, 359
  - nppiLn\_32f\_C1IR, 360
  - nppiLn\_32f\_C1R, 360
  - nppiLn\_32f\_C3IR, 360
  - nppiLn\_32f\_C3R, 361
  - nppiLn\_8u\_C1IRSfs, 361
  - nppiLn\_8u\_C1RSfs, 361
  - nppiLn\_8u\_C3IRSfs, 362
  - nppiLn\_8u\_C3RSfs, 362
- image\_lshifc
  - nppiLShiftC\_16u\_AC4IR, 423
  - nppiLShiftC\_16u\_AC4R, 423
  - nppiLShiftC\_16u\_C1IR, 423
  - nppiLShiftC\_16u\_C1R, 424
  - nppiLShiftC\_16u\_C3IR, 424
  - nppiLShiftC\_16u\_C3R, 424
  - nppiLShiftC\_16u\_C4IR, 425
  - nppiLShiftC\_16u\_C4R, 425
  - nppiLShiftC\_32s\_AC4IR, 425
  - nppiLShiftC\_32s\_AC4R, 426
  - nppiLShiftC\_32s\_C1IR, 426
  - nppiLShiftC\_32s\_C1R, 426
  - nppiLShiftC\_32s\_C3IR, 427
  - nppiLShiftC\_32s\_C3R, 427
  - nppiLShiftC\_32s\_C4IR, 427
  - nppiLShiftC\_32s\_C4R, 428
  - nppiLShiftC\_8u\_AC4IR, 428
  - nppiLShiftC\_8u\_AC4R, 428
  - nppiLShiftC\_8u\_C1IR, 429
  - nppiLShiftC\_8u\_C1R, 429
  - nppiLShiftC\_8u\_C3IR, 429
  - nppiLShiftC\_8u\_C3R, 430
  - nppiLShiftC\_8u\_C4IR, 430
  - nppiLShiftC\_8u\_C4R, 430
- image\_max
  - nppiMax\_16s\_AC4R, 1504
  - nppiMax\_16s\_C1R, 1504
  - nppiMax\_16s\_C3R, 1505
  - nppiMax\_16s\_C4R, 1505
  - nppiMax\_16u\_AC4R, 1505
  - nppiMax\_16u\_C1R, 1506
  - nppiMax\_16u\_C3R, 1506
  - nppiMax\_16u\_C4R, 1507
  - nppiMax\_32f\_AC4R, 1507
  - nppiMax\_32f\_C1R, 1507
  - nppiMax\_32f\_C3R, 1508
  - nppiMax\_32f\_C4R, 1508
  - nppiMax\_8u\_AC4R, 1508
  - nppiMax\_8u\_C1R, 1509
  - nppiMax\_8u\_C3R, 1509
  - nppiMax\_8u\_C4R, 1510
  - nppiMaxGetBufferHostSize\_16s\_AC4R, 1510
  - nppiMaxGetBufferHostSize\_16s\_C1R, 1510
  - nppiMaxGetBufferHostSize\_16s\_C3R, 1510
  - nppiMaxGetBufferHostSize\_16s\_C4R, 1511
  - nppiMaxGetBufferHostSize\_16u\_AC4R, 1511
  - nppiMaxGetBufferHostSize\_16u\_C1R, 1511
  - nppiMaxGetBufferHostSize\_16u\_C3R, 1512
  - nppiMaxGetBufferHostSize\_16u\_C4R, 1512
  - nppiMaxGetBufferHostSize\_32f\_AC4R, 1512
  - nppiMaxGetBufferHostSize\_32f\_C1R, 1512
  - nppiMaxGetBufferHostSize\_32f\_C3R, 1513
  - nppiMaxGetBufferHostSize\_32f\_C4R, 1513
  - nppiMaxGetBufferHostSize\_8u\_AC4R, 1513
  - nppiMaxGetBufferHostSize\_8u\_C1R, 1514
  - nppiMaxGetBufferHostSize\_8u\_C3R, 1514
  - nppiMaxGetBufferHostSize\_8u\_C4R, 1514
- image\_max\_index
  - nppiMaxIndx\_16s\_AC4R, 1517
  - nppiMaxIndx\_16s\_C1R, 1518
  - nppiMaxIndx\_16s\_C3R, 1518
  - nppiMaxIndx\_16s\_C4R, 1518
  - nppiMaxIndx\_16u\_AC4R, 1519
  - nppiMaxIndx\_16u\_C1R, 1519
  - nppiMaxIndx\_16u\_C3R, 1520
  - nppiMaxIndx\_16u\_C4R, 1520
  - nppiMaxIndx\_32f\_AC4R, 1520
  - nppiMaxIndx\_32f\_C1R, 1521
  - nppiMaxIndx\_32f\_C3R, 1521
  - nppiMaxIndx\_32f\_C4R, 1522
  - nppiMaxIndx\_8u\_AC4R, 1522
  - nppiMaxIndx\_8u\_C1R, 1522

- nppiMaxIndx\_8u\_C3R, [1523](#)
- nppiMaxIndx\_8u\_C4R, [1523](#)
- nppiMaxIndxGetBufferHostSize\_16s\_AC4R, [1524](#)
- nppiMaxIndxGetBufferHostSize\_16s\_C1R, [1524](#)
- nppiMaxIndxGetBufferHostSize\_16s\_C3R, [1524](#)
- nppiMaxIndxGetBufferHostSize\_16s\_C4R, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_AC4R, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_C1R, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_C3R, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_C4R, [1526](#)
- nppiMaxIndxGetBufferHostSize\_32f\_AC4R, [1526](#)
- nppiMaxIndxGetBufferHostSize\_32f\_C1R, [1526](#)
- nppiMaxIndxGetBufferHostSize\_32f\_C3R, [1527](#)
- nppiMaxIndxGetBufferHostSize\_32f\_C4R, [1527](#)
- nppiMaxIndxGetBufferHostSize\_8u\_AC4R, [1527](#)
- nppiMaxIndxGetBufferHostSize\_8u\_C1R, [1527](#)
- nppiMaxIndxGetBufferHostSize\_8u\_C3R, [1528](#)
- nppiMaxIndxGetBufferHostSize\_8u\_C4R, [1528](#)
- image\_maxevery
  - nppiMaxEvery\_16s\_AC4IR, [1833](#)
  - nppiMaxEvery\_16s\_C1IR, [1833](#)
  - nppiMaxEvery\_16s\_C3IR, [1834](#)
  - nppiMaxEvery\_16s\_C4IR, [1834](#)
  - nppiMaxEvery\_16u\_AC4IR, [1834](#)
  - nppiMaxEvery\_16u\_C1IR, [1835](#)
  - nppiMaxEvery\_16u\_C3IR, [1835](#)
  - nppiMaxEvery\_16u\_C4IR, [1835](#)
  - nppiMaxEvery\_32f\_AC4IR, [1836](#)
  - nppiMaxEvery\_32f\_C1IR, [1836](#)
  - nppiMaxEvery\_32f\_C3IR, [1836](#)
  - nppiMaxEvery\_32f\_C4IR, [1837](#)
  - nppiMaxEvery\_8u\_AC4IR, [1837](#)
  - nppiMaxEvery\_8u\_C1IR, [1837](#)
  - nppiMaxEvery\_8u\_C3IR, [1838](#)
  - nppiMaxEvery\_8u\_C4IR, [1838](#)
- image\_maximum\_error
  - nppiMaximumError\_16s\_C1R, [2027](#)
  - nppiMaximumError\_16s\_C2R, [2028](#)
  - nppiMaximumError\_16s\_C3R, [2028](#)
  - nppiMaximumError\_16s\_C4R, [2028](#)
  - nppiMaximumError\_16sc\_C1R, [2029](#)
  - nppiMaximumError\_16sc\_C2R, [2029](#)
  - nppiMaximumError\_16sc\_C3R, [2030](#)
  - nppiMaximumError\_16sc\_C4R, [2030](#)
  - nppiMaximumError\_16u\_C1R, [2031](#)
  - nppiMaximumError\_16u\_C2R, [2031](#)
  - nppiMaximumError\_16u\_C3R, [2031](#)
  - nppiMaximumError\_16u\_C4R, [2032](#)
  - nppiMaximumError\_32f\_C1R, [2032](#)
  - nppiMaximumError\_32f\_C2R, [2033](#)
  - nppiMaximumError\_32f\_C3R, [2033](#)
  - nppiMaximumError\_32f\_C4R, [2034](#)
  - nppiMaximumError\_32fc\_C1R, [2034](#)
  - nppiMaximumError\_32fc\_C2R, [2035](#)
  - nppiMaximumError\_32fc\_C3R, [2035](#)
  - nppiMaximumError\_32fc\_C4R, [2035](#)
  - nppiMaximumError\_32s\_C1R, [2036](#)
  - nppiMaximumError\_32s\_C2R, [2036](#)
  - nppiMaximumError\_32s\_C3R, [2037](#)
  - nppiMaximumError\_32s\_C4R, [2037](#)
  - nppiMaximumError\_32sc\_C1R, [2038](#)
  - nppiMaximumError\_32sc\_C2R, [2038](#)
  - nppiMaximumError\_32sc\_C3R, [2038](#)
  - nppiMaximumError\_32sc\_C4R, [2039](#)
  - nppiMaximumError\_32u\_C1R, [2039](#)
  - nppiMaximumError\_32u\_C2R, [2040](#)
  - nppiMaximumError\_32u\_C3R, [2040](#)
  - nppiMaximumError\_32u\_C4R, [2041](#)
  - nppiMaximumError\_64f\_C1R, [2041](#)
  - nppiMaximumError\_64f\_C2R, [2041](#)
  - nppiMaximumError\_64f\_C3R, [2042](#)
  - nppiMaximumError\_64f\_C4R, [2042](#)
  - nppiMaximumError\_8s\_C1R, [2043](#)
  - nppiMaximumError\_8s\_C2R, [2043](#)
  - nppiMaximumError\_8s\_C3R, [2044](#)
  - nppiMaximumError\_8s\_C4R, [2044](#)
  - nppiMaximumError\_8u\_C1R, [2044](#)
  - nppiMaximumError\_8u\_C2R, [2045](#)
  - nppiMaximumError\_8u\_C3R, [2045](#)
  - nppiMaximumError\_8u\_C4R, [2046](#)
- image\_maximum\_relative\_error
  - nppiMaximumRelativeError\_16s\_C1R, [2073](#)
  - nppiMaximumRelativeError\_16s\_C2R, [2074](#)
  - nppiMaximumRelativeError\_16s\_C3R, [2074](#)
  - nppiMaximumRelativeError\_16s\_C4R, [2075](#)
  - nppiMaximumRelativeError\_16sc\_C1R, [2075](#)
  - nppiMaximumRelativeError\_16sc\_C2R, [2076](#)
  - nppiMaximumRelativeError\_16sc\_C3R, [2076](#)
  - nppiMaximumRelativeError\_16sc\_C4R, [2076](#)
  - nppiMaximumRelativeError\_16u\_C1R, [2077](#)
  - nppiMaximumRelativeError\_16u\_C2R, [2077](#)
  - nppiMaximumRelativeError\_16u\_C3R, [2078](#)



- nppiMaximumRelativeError\_16u\_C4R, 2078
- nppiMaximumRelativeError\_32f\_C1R, 2079
- nppiMaximumRelativeError\_32f\_C2R, 2079
- nppiMaximumRelativeError\_32f\_C3R, 2080
- nppiMaximumRelativeError\_32f\_C4R, 2080
- nppiMaximumRelativeError\_32fc\_C1R, 2081
- nppiMaximumRelativeError\_32fc\_C2R, 2081
- nppiMaximumRelativeError\_32fc\_C3R, 2081
- nppiMaximumRelativeError\_32fc\_C4R, 2082
- nppiMaximumRelativeError\_32s\_C1R, 2082
- nppiMaximumRelativeError\_32s\_C2R, 2083
- nppiMaximumRelativeError\_32s\_C3R, 2083
- nppiMaximumRelativeError\_32s\_C4R, 2084
- nppiMaximumRelativeError\_32sc\_C1R, 2084
- nppiMaximumRelativeError\_32sc\_C2R, 2085
- nppiMaximumRelativeError\_32sc\_C3R, 2085
- nppiMaximumRelativeError\_32sc\_C4R, 2086
- nppiMaximumRelativeError\_32u\_C1R, 2086
- nppiMaximumRelativeError\_32u\_C2R, 2086
- nppiMaximumRelativeError\_32u\_C3R, 2087
- nppiMaximumRelativeError\_32u\_C4R, 2087
- nppiMaximumRelativeError\_64f\_C1R, 2088
- nppiMaximumRelativeError\_64f\_C2R, 2088
- nppiMaximumRelativeError\_64f\_C3R, 2089
- nppiMaximumRelativeError\_64f\_C4R, 2089
- nppiMaximumRelativeError\_8s\_C1R, 2090
- nppiMaximumRelativeError\_8s\_C2R, 2090
- nppiMaximumRelativeError\_8s\_C3R, 2091
- nppiMaximumRelativeError\_8s\_C4R, 2091
- nppiMaximumRelativeError\_8u\_C1R, 2091
- nppiMaximumRelativeError\_8u\_C2R, 2092
- nppiMaximumRelativeError\_8u\_C3R, 2092
- nppiMaximumRelativeError\_8u\_C4R, 2093
- image\_mean
  - nppiMean\_16s\_AC4R, 1564
  - nppiMean\_16s\_C1R, 1564
  - nppiMean\_16s\_C3R, 1564
  - nppiMean\_16s\_C4R, 1565
  - nppiMean\_16u\_AC4R, 1565
  - nppiMean\_16u\_C1MR, 1565
  - nppiMean\_16u\_C1R, 1566
  - nppiMean\_16u\_C3CMR, 1566
  - nppiMean\_16u\_C3R, 1566
  - nppiMean\_16u\_C4R, 1567
  - nppiMean\_32f\_AC4R, 1567
  - nppiMean\_32f\_C1MR, 1568
  - nppiMean\_32f\_C1R, 1568
  - nppiMean\_32f\_C3CMR, 1568
  - nppiMean\_32f\_C3R, 1569
  - nppiMean\_32f\_C4R, 1569
  - nppiMean\_8s\_C1MR, 1570
  - nppiMean\_8s\_C3CMR, 1570
  - nppiMean\_8u\_AC4R, 1571
  - nppiMean\_8u\_C1MR, 1571
  - nppiMean\_8u\_C1R, 1571
  - nppiMean\_8u\_C3CMR, 1572
  - nppiMean\_8u\_C3R, 1572
  - nppiMean\_8u\_C4R, 1573
  - nppiMeanGetBufferHostSize\_16s\_AC4R, 1573
  - nppiMeanGetBufferHostSize\_16s\_C1R, 1573
  - nppiMeanGetBufferHostSize\_16s\_C3R, 1574
  - nppiMeanGetBufferHostSize\_16s\_C4R, 1574
  - nppiMeanGetBufferHostSize\_16u\_AC4R, 1574
  - nppiMeanGetBufferHostSize\_16u\_C1MR, 1574
  - nppiMeanGetBufferHostSize\_16u\_C1R, 1575
  - nppiMeanGetBufferHostSize\_16u\_C3CMR, 1575
  - nppiMeanGetBufferHostSize\_16u\_C3R, 1575
  - nppiMeanGetBufferHostSize\_16u\_C4R, 1576
  - nppiMeanGetBufferHostSize\_32f\_AC4R, 1576
  - nppiMeanGetBufferHostSize\_32f\_C1MR, 1576
  - nppiMeanGetBufferHostSize\_32f\_C1R, 1576
  - nppiMeanGetBufferHostSize\_32f\_C3CMR, 1577
  - nppiMeanGetBufferHostSize\_32f\_C3R, 1577
  - nppiMeanGetBufferHostSize\_32f\_C4R, 1577
  - nppiMeanGetBufferHostSize\_8s\_C1MR, 1578
  - nppiMeanGetBufferHostSize\_8s\_C3CMR, 1578
  - nppiMeanGetBufferHostSize\_8u\_AC4R, 1578
  - nppiMeanGetBufferHostSize\_8u\_C1MR, 1578
  - nppiMeanGetBufferHostSize\_8u\_C1R, 1579
  - nppiMeanGetBufferHostSize\_8u\_C3CMR, 1579
  - nppiMeanGetBufferHostSize\_8u\_C3R, 1579
  - nppiMeanGetBufferHostSize\_8u\_C4R, 1580
- image\_mean\_stddev
  - nppiMean\_StdDev\_16u\_C1MR, 1584
  - nppiMean\_StdDev\_16u\_C1R, 1584
  - nppiMean\_StdDev\_16u\_C3CMR, 1585
  - nppiMean\_StdDev\_16u\_C3CR, 1585
  - nppiMean\_StdDev\_32f\_C1MR, 1586
  - nppiMean\_StdDev\_32f\_C1R, 1586
  - nppiMean\_StdDev\_32f\_C3CMR, 1587
  - nppiMean\_StdDev\_32f\_C3CR, 1587
  - nppiMean\_StdDev\_8s\_C1MR, 1588
  - nppiMean\_StdDev\_8s\_C1R, 1588
  - nppiMean\_StdDev\_8s\_C3CMR, 1589
  - nppiMean\_StdDev\_8s\_C3CR, 1589
  - nppiMean\_StdDev\_8u\_C1MR, 1590
  - nppiMean\_StdDev\_8u\_C1R, 1590

- nppiMean\_StdDev\_8u\_C3CMR, [1591](#)
- nppiMean\_StdDev\_8u\_C3CR, [1591](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_-C1MR, [1592](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_-C1R, [1592](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_-C3CMR, [1592](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_-C3CR, [1593](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_-C1MR, [1593](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_-C1R, [1593](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_-C3CMR, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_-C3CR, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_-C1MR, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_-C1R, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_-C3CMR, [1595](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_-C3CR, [1595](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_-C1MR, [1595](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_-C1R, [1596](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_-C3CMR, [1596](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_-C3CR, [1596](#)
- image\_memory\_management
  - nppiFree, [2120](#)
  - nppiMalloc\_16s\_C1, [2120](#)
  - nppiMalloc\_16s\_C2, [2120](#)
  - nppiMalloc\_16s\_C4, [2121](#)
  - nppiMalloc\_16sc\_C1, [2121](#)
  - nppiMalloc\_16sc\_C2, [2121](#)
  - nppiMalloc\_16sc\_C3, [2122](#)
  - nppiMalloc\_16sc\_C4, [2122](#)
  - nppiMalloc\_16u\_C1, [2122](#)
  - nppiMalloc\_16u\_C2, [2122](#)
  - nppiMalloc\_16u\_C3, [2123](#)
  - nppiMalloc\_16u\_C4, [2123](#)
  - nppiMalloc\_32f\_C1, [2123](#)
  - nppiMalloc\_32f\_C2, [2124](#)
  - nppiMalloc\_32f\_C3, [2124](#)
  - nppiMalloc\_32f\_C4, [2124](#)
  - nppiMalloc\_32fc\_C1, [2124](#)
  - nppiMalloc\_32fc\_C2, [2125](#)
  - nppiMalloc\_32fc\_C3, [2125](#)
  - nppiMalloc\_32fc\_C4, [2125](#)
  - nppiMalloc\_32s\_C1, [2126](#)
  - nppiMalloc\_32s\_C3, [2126](#)
  - nppiMalloc\_32s\_C4, [2126](#)
  - nppiMalloc\_32sc\_C1, [2126](#)
  - nppiMalloc\_32sc\_C2, [2127](#)
  - nppiMalloc\_32sc\_C3, [2127](#)
  - nppiMalloc\_32sc\_C4, [2127](#)
  - nppiMalloc\_8u\_C1, [2128](#)
  - nppiMalloc\_8u\_C2, [2128](#)
  - nppiMalloc\_8u\_C3, [2128](#)
  - nppiMalloc\_8u\_C4, [2128](#)
- image\_min
  - nppiMin\_16s\_AC4R, [1477](#)
  - nppiMin\_16s\_C1R, [1477](#)
  - nppiMin\_16s\_C3R, [1478](#)
  - nppiMin\_16s\_C4R, [1478](#)
  - nppiMin\_16u\_AC4R, [1478](#)
  - nppiMin\_16u\_C1R, [1479](#)
  - nppiMin\_16u\_C3R, [1479](#)
  - nppiMin\_16u\_C4R, [1480](#)
  - nppiMin\_32f\_AC4R, [1480](#)
  - nppiMin\_32f\_C1R, [1480](#)
  - nppiMin\_32f\_C3R, [1481](#)
  - nppiMin\_32f\_C4R, [1481](#)
  - nppiMin\_8u\_AC4R, [1481](#)
  - nppiMin\_8u\_C1R, [1482](#)
  - nppiMin\_8u\_C3R, [1482](#)
  - nppiMin\_8u\_C4R, [1483](#)
  - nppiMinGetBufferHostSize\_16s\_AC4R, [1483](#)
  - nppiMinGetBufferHostSize\_16s\_C1R, [1483](#)
  - nppiMinGetBufferHostSize\_16s\_C3R, [1483](#)
  - nppiMinGetBufferHostSize\_16s\_C4R, [1484](#)
  - nppiMinGetBufferHostSize\_16u\_AC4R, [1484](#)
  - nppiMinGetBufferHostSize\_16u\_C1R, [1484](#)
  - nppiMinGetBufferHostSize\_16u\_C3R, [1485](#)
  - nppiMinGetBufferHostSize\_16u\_C4R, [1485](#)
  - nppiMinGetBufferHostSize\_32f\_AC4R, [1485](#)
  - nppiMinGetBufferHostSize\_32f\_C1R, [1485](#)
  - nppiMinGetBufferHostSize\_32f\_C3R, [1486](#)
  - nppiMinGetBufferHostSize\_32f\_C4R, [1486](#)
  - nppiMinGetBufferHostSize\_8u\_AC4R, [1486](#)
  - nppiMinGetBufferHostSize\_8u\_C1R, [1487](#)
  - nppiMinGetBufferHostSize\_8u\_C3R, [1487](#)
  - nppiMinGetBufferHostSize\_8u\_C4R, [1487](#)
- image\_min\_index
  - nppiMinIndx\_16s\_AC4R, [1490](#)
  - nppiMinIndx\_16s\_C1R, [1491](#)
  - nppiMinIndx\_16s\_C3R, [1491](#)
  - nppiMinIndx\_16s\_C4R, [1491](#)
  - nppiMinIndx\_16u\_AC4R, [1492](#)
  - nppiMinIndx\_16u\_C1R, [1492](#)
  - nppiMinIndx\_16u\_C3R, [1493](#)
  - nppiMinIndx\_16u\_C4R, [1493](#)

- nppiMinIndx\_32f\_AC4R, [1493](#)
- nppiMinIndx\_32f\_C1R, [1494](#)
- nppiMinIndx\_32f\_C3R, [1494](#)
- nppiMinIndx\_32f\_C4R, [1495](#)
- nppiMinIndx\_8u\_AC4R, [1495](#)
- nppiMinIndx\_8u\_C1R, [1495](#)
- nppiMinIndx\_8u\_C3R, [1496](#)
- nppiMinIndx\_8u\_C4R, [1496](#)
- nppiMinIndxGetBufferHostSize\_16s\_AC4R, [1497](#)
- nppiMinIndxGetBufferHostSize\_16s\_C1R, [1497](#)
- nppiMinIndxGetBufferHostSize\_16s\_C3R, [1497](#)
- nppiMinIndxGetBufferHostSize\_16s\_C4R, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_AC4R, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_C1R, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_C3R, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_C4R, [1499](#)
- nppiMinIndxGetBufferHostSize\_32f\_AC4R, [1499](#)
- nppiMinIndxGetBufferHostSize\_32f\_C1R, [1499](#)
- nppiMinIndxGetBufferHostSize\_32f\_C3R, [1500](#)
- nppiMinIndxGetBufferHostSize\_32f\_C4R, [1500](#)
- nppiMinIndxGetBufferHostSize\_8u\_AC4R, [1500](#)
- nppiMinIndxGetBufferHostSize\_8u\_C1R, [1500](#)
- nppiMinIndxGetBufferHostSize\_8u\_C3R, [1501](#)
- nppiMinIndxGetBufferHostSize\_8u\_C4R, [1501](#)
- image\_min\_max
  - nppiMinMax\_16s\_AC4R, [1531](#)
  - nppiMinMax\_16s\_C1R, [1531](#)
  - nppiMinMax\_16s\_C3R, [1532](#)
  - nppiMinMax\_16s\_C4R, [1532](#)
  - nppiMinMax\_16u\_AC4R, [1533](#)
  - nppiMinMax\_16u\_C1R, [1533](#)
  - nppiMinMax\_16u\_C3R, [1533](#)
  - nppiMinMax\_16u\_C4R, [1534](#)
  - nppiMinMax\_32f\_AC4R, [1534](#)
  - nppiMinMax\_32f\_C1R, [1535](#)
  - nppiMinMax\_32f\_C3R, [1535](#)
  - nppiMinMax\_32f\_C4R, [1535](#)
  - nppiMinMax\_8u\_AC4R, [1536](#)
  - nppiMinMax\_8u\_C1R, [1536](#)
  - nppiMinMax\_8u\_C3R, [1537](#)
  - nppiMinMax\_8u\_C4R, [1537](#)
  - nppiMinMaxGetBufferHostSize\_16s\_AC4R, [1537](#)
  - nppiMinMaxGetBufferHostSize\_16s\_C1R, [1538](#)
  - nppiMinMaxGetBufferHostSize\_16s\_C3R, [1538](#)
  - nppiMinMaxGetBufferHostSize\_16s\_C4R, [1538](#)
  - nppiMinMaxGetBufferHostSize\_16u\_AC4R, [1539](#)
  - nppiMinMaxGetBufferHostSize\_16u\_C1R, [1539](#)
  - nppiMinMaxGetBufferHostSize\_16u\_C3R, [1539](#)
  - nppiMinMaxGetBufferHostSize\_16u\_C4R, [1539](#)
  - nppiMinMaxGetBufferHostSize\_32f\_AC4R, [1540](#)
  - nppiMinMaxGetBufferHostSize\_32f\_C1R, [1540](#)
  - nppiMinMaxGetBufferHostSize\_32f\_C3R, [1540](#)
  - nppiMinMaxGetBufferHostSize\_32f\_C4R, [1541](#)
  - nppiMinMaxGetBufferHostSize\_8u\_AC4R, [1541](#)
  - nppiMinMaxGetBufferHostSize\_8u\_C1R, [1541](#)
  - nppiMinMaxGetBufferHostSize\_8u\_C3R, [1541](#)
  - nppiMinMaxGetBufferHostSize\_8u\_C4R, [1542](#)
- image\_min\_max\_index
  - nppiMinMaxIndx\_16u\_C1MR, [1546](#)
  - nppiMinMaxIndx\_16u\_C1R, [1547](#)
  - nppiMinMaxIndx\_16u\_C3CMR, [1547](#)
  - nppiMinMaxIndx\_16u\_C3CR, [1548](#)
  - nppiMinMaxIndx\_32f\_C1MR, [1548](#)
  - nppiMinMaxIndx\_32f\_C1R, [1549](#)
  - nppiMinMaxIndx\_32f\_C3CMR, [1549](#)
  - nppiMinMaxIndx\_32f\_C3CR, [1550](#)
  - nppiMinMaxIndx\_8s\_C1MR, [1551](#)
  - nppiMinMaxIndx\_8s\_C1R, [1551](#)
  - nppiMinMaxIndx\_8s\_C3CMR, [1552](#)
  - nppiMinMaxIndx\_8s\_C3CR, [1552](#)
  - nppiMinMaxIndx\_8u\_C1MR, [1553](#)
  - nppiMinMaxIndx\_8u\_C1R, [1553](#)
  - nppiMinMaxIndx\_8u\_C3CMR, [1554](#)
  - nppiMinMaxIndx\_8u\_C3CR, [1554](#)
  - nppiMinMaxIndxGetBufferHostSize\_16u\_C1MR, [1555](#)



- nppiMinMaxIdxGetBufferHostSize\_16u\_-C1R, [1555](#)
- nppiMinMaxIdxGetBufferHostSize\_16u\_-C3CMR, [1555](#)
- nppiMinMaxIdxGetBufferHostSize\_16u\_-C3CR, [1556](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_-C1MR, [1556](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_-C1R, [1556](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_-C3CMR, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_-C3CR, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_-C1MR, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_C1R, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_-C3CMR, [1558](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_-C3CR, [1558](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_-C1MR, [1558](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_-C1R, [1559](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_-C3CMR, [1559](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_-C3CR, [1559](#)
- image\_minevery
  - nppiMinEvery\_16s\_AC4IR, [1840](#)
  - nppiMinEvery\_16s\_C1IR, [1840](#)
  - nppiMinEvery\_16s\_C3IR, [1841](#)
  - nppiMinEvery\_16s\_C4IR, [1841](#)
  - nppiMinEvery\_16u\_AC4IR, [1841](#)
  - nppiMinEvery\_16u\_C1IR, [1842](#)
  - nppiMinEvery\_16u\_C3IR, [1842](#)
  - nppiMinEvery\_16u\_C4IR, [1842](#)
  - nppiMinEvery\_32f\_AC4IR, [1843](#)
  - nppiMinEvery\_32f\_C1IR, [1843](#)
  - nppiMinEvery\_32f\_C3IR, [1843](#)
  - nppiMinEvery\_32f\_C4IR, [1844](#)
  - nppiMinEvery\_8u\_AC4IR, [1844](#)
  - nppiMinEvery\_8u\_C1IR, [1844](#)
  - nppiMinEvery\_8u\_C3IR, [1845](#)
  - nppiMinEvery\_8u\_C4IR, [1845](#)
- image\_mirror
  - nppiMirror\_16s\_AC4IR, [1253](#)
  - nppiMirror\_16s\_AC4R, [1253](#)
  - nppiMirror\_16s\_C1IR, [1254](#)
  - nppiMirror\_16s\_C1R, [1254](#)
  - nppiMirror\_16s\_C3IR, [1254](#)
  - nppiMirror\_16s\_C3R, [1255](#)
  - nppiMirror\_16s\_C4IR, [1255](#)
  - nppiMirror\_16s\_C4R, [1255](#)
  - nppiMirror\_16u\_AC4IR, [1256](#)
  - nppiMirror\_16u\_AC4R, [1256](#)
  - nppiMirror\_16u\_C1IR, [1256](#)
  - nppiMirror\_16u\_C1R, [1257](#)
  - nppiMirror\_16u\_C3IR, [1257](#)
  - nppiMirror\_16u\_C3R, [1257](#)
  - nppiMirror\_16u\_C4IR, [1258](#)
  - nppiMirror\_16u\_C4R, [1258](#)
  - nppiMirror\_32f\_AC4IR, [1258](#)
  - nppiMirror\_32f\_AC4R, [1259](#)
  - nppiMirror\_32f\_C1IR, [1259](#)
  - nppiMirror\_32f\_C1R, [1259](#)
  - nppiMirror\_32f\_C3IR, [1260](#)
  - nppiMirror\_32f\_C3R, [1260](#)
  - nppiMirror\_32f\_C4IR, [1260](#)
  - nppiMirror\_32f\_C4R, [1261](#)
  - nppiMirror\_32s\_AC4IR, [1261](#)
  - nppiMirror\_32s\_AC4R, [1261](#)
  - nppiMirror\_32s\_C1IR, [1262](#)
  - nppiMirror\_32s\_C1R, [1262](#)
  - nppiMirror\_32s\_C3IR, [1262](#)
  - nppiMirror\_32s\_C3R, [1263](#)
  - nppiMirror\_32s\_C4IR, [1263](#)
  - nppiMirror\_32s\_C4R, [1263](#)
  - nppiMirror\_8u\_AC4IR, [1264](#)
  - nppiMirror\_8u\_AC4R, [1264](#)
  - nppiMirror\_8u\_C1IR, [1264](#)
  - nppiMirror\_8u\_C1R, [1265](#)
  - nppiMirror\_8u\_C3IR, [1265](#)
  - nppiMirror\_8u\_C3R, [1265](#)
  - nppiMirror\_8u\_C4IR, [1266](#)
  - nppiMirror\_8u\_C4R, [1266](#)
- image\_mul
  - nppiMul\_16s\_AC4IRSfs, [213](#)
  - nppiMul\_16s\_AC4RSfs, [213](#)
  - nppiMul\_16s\_C1IRSfs, [214](#)
  - nppiMul\_16s\_C1RSfs, [214](#)
  - nppiMul\_16s\_C3IRSfs, [215](#)
  - nppiMul\_16s\_C3RSfs, [215](#)
  - nppiMul\_16s\_C4IRSfs, [215](#)
  - nppiMul\_16s\_C4RSfs, [216](#)
  - nppiMul\_16sc\_AC4IRSfs, [216](#)
  - nppiMul\_16sc\_AC4RSfs, [217](#)
  - nppiMul\_16sc\_C1IRSfs, [217](#)
  - nppiMul\_16sc\_C1RSfs, [217](#)
  - nppiMul\_16sc\_C3IRSfs, [218](#)
  - nppiMul\_16sc\_C3RSfs, [218](#)
  - nppiMul\_16u\_AC4IRSfs, [219](#)
  - nppiMul\_16u\_AC4RSfs, [219](#)
  - nppiMul\_16u\_C1IRSfs, [220](#)
  - nppiMul\_16u\_C1RSfs, [220](#)
  - nppiMul\_16u\_C3IRSfs, [220](#)

- nppiMul\_16u\_C3RSfs, [221](#)
- nppiMul\_16u\_C4IRSfs, [221](#)
- nppiMul\_16u\_C4RSfs, [222](#)
- nppiMul\_32f\_AC4IR, [222](#)
- nppiMul\_32f\_AC4R, [222](#)
- nppiMul\_32f\_C1IR, [223](#)
- nppiMul\_32f\_C1R, [223](#)
- nppiMul\_32f\_C3IR, [224](#)
- nppiMul\_32f\_C3R, [224](#)
- nppiMul\_32f\_C4IR, [224](#)
- nppiMul\_32f\_C4R, [225](#)
- nppiMul\_32fc\_AC4IR, [225](#)
- nppiMul\_32fc\_AC4R, [225](#)
- nppiMul\_32fc\_C1IR, [226](#)
- nppiMul\_32fc\_C1R, [226](#)
- nppiMul\_32fc\_C3IR, [227](#)
- nppiMul\_32fc\_C3R, [227](#)
- nppiMul\_32fc\_C4IR, [227](#)
- nppiMul\_32fc\_C4R, [228](#)
- nppiMul\_32s\_C1IRSfs, [228](#)
- nppiMul\_32s\_C1R, [229](#)
- nppiMul\_32s\_C1RSfs, [229](#)
- nppiMul\_32s\_C3IRSfs, [229](#)
- nppiMul\_32s\_C3RSfs, [230](#)
- nppiMul\_32sc\_AC4IRSfs, [230](#)
- nppiMul\_32sc\_AC4RSfs, [231](#)
- nppiMul\_32sc\_C1IRSfs, [231](#)
- nppiMul\_32sc\_C1RSfs, [231](#)
- nppiMul\_32sc\_C3IRSfs, [232](#)
- nppiMul\_32sc\_C3RSfs, [232](#)
- nppiMul\_8u\_AC4IRSfs, [233](#)
- nppiMul\_8u\_AC4RSfs, [233](#)
- nppiMul\_8u\_C1IRSfs, [234](#)
- nppiMul\_8u\_C1RSfs, [234](#)
- nppiMul\_8u\_C3IRSfs, [234](#)
- nppiMul\_8u\_C3RSfs, [235](#)
- nppiMul\_8u\_C4IRSfs, [235](#)
- nppiMul\_8u\_C4RSfs, [236](#)
- image\_mulc
  - nppiMulC\_16s\_AC4IRSfs, [86](#)
  - nppiMulC\_16s\_AC4RSfs, [86](#)
  - nppiMulC\_16s\_C1IRSfs, [86](#)
  - nppiMulC\_16s\_C1RSfs, [87](#)
  - nppiMulC\_16s\_C3IRSfs, [87](#)
  - nppiMulC\_16s\_C3RSfs, [87](#)
  - nppiMulC\_16s\_C4IRSfs, [88](#)
  - nppiMulC\_16s\_C4RSfs, [88](#)
  - nppiMulC\_16sc\_AC4IRSfs, [89](#)
  - nppiMulC\_16sc\_AC4RSfs, [89](#)
  - nppiMulC\_16sc\_C1IRSfs, [89](#)
  - nppiMulC\_16sc\_C1RSfs, [90](#)
  - nppiMulC\_16sc\_C3IRSfs, [90](#)
  - nppiMulC\_16sc\_C3RSfs, [91](#)
  - nppiMulC\_16u\_AC4IRSfs, [91](#)
- nppiMulC\_16u\_AC4RSfs, [91](#)
- nppiMulC\_16u\_C1IRSfs, [92](#)
- nppiMulC\_16u\_C1RSfs, [92](#)
- nppiMulC\_16u\_C3IRSfs, [93](#)
- nppiMulC\_16u\_C3RSfs, [93](#)
- nppiMulC\_16u\_C4IRSfs, [93](#)
- nppiMulC\_16u\_C4RSfs, [94](#)
- nppiMulC\_32f\_AC4IR, [94](#)
- nppiMulC\_32f\_AC4R, [94](#)
- nppiMulC\_32f\_C1IR, [95](#)
- nppiMulC\_32f\_C1R, [95](#)
- nppiMulC\_32f\_C3IR, [95](#)
- nppiMulC\_32f\_C3R, [96](#)
- nppiMulC\_32f\_C4IR, [96](#)
- nppiMulC\_32f\_C4R, [96](#)
- nppiMulC\_32fc\_AC4IR, [97](#)
- nppiMulC\_32fc\_AC4R, [97](#)
- nppiMulC\_32fc\_C1IR, [97](#)
- nppiMulC\_32fc\_C1R, [98](#)
- nppiMulC\_32fc\_C3IR, [98](#)
- nppiMulC\_32fc\_C3R, [98](#)
- nppiMulC\_32fc\_C4IR, [99](#)
- nppiMulC\_32fc\_C4R, [99](#)
- nppiMulC\_32s\_C1IRSfs, [100](#)
- nppiMulC\_32s\_C1RSfs, [100](#)
- nppiMulC\_32s\_C3IRSfs, [100](#)
- nppiMulC\_32s\_C3RSfs, [101](#)
- nppiMulC\_32sc\_AC4IRSfs, [101](#)
- nppiMulC\_32sc\_AC4RSfs, [101](#)
- nppiMulC\_32sc\_C1IRSfs, [102](#)
- nppiMulC\_32sc\_C1RSfs, [102](#)
- nppiMulC\_32sc\_C3IRSfs, [103](#)
- nppiMulC\_32sc\_C3RSfs, [103](#)
- nppiMulC\_8u\_AC4IRSfs, [103](#)
- nppiMulC\_8u\_AC4RSfs, [104](#)
- nppiMulC\_8u\_C1IRSfs, [104](#)
- nppiMulC\_8u\_C1RSfs, [105](#)
- nppiMulC\_8u\_C3IRSfs, [105](#)
- nppiMulC\_8u\_C3RSfs, [105](#)
- nppiMulC\_8u\_C4IRSfs, [106](#)
- nppiMulC\_8u\_C4RSfs, [106](#)
- image\_mulcscale
  - nppiMulCScale\_16u\_AC4IR, [108](#)
  - nppiMulCScale\_16u\_AC4R, [108](#)
  - nppiMulCScale\_16u\_C1IR, [109](#)
  - nppiMulCScale\_16u\_C1R, [109](#)
  - nppiMulCScale\_16u\_C3IR, [109](#)
  - nppiMulCScale\_16u\_C3R, [110](#)
  - nppiMulCScale\_16u\_C4IR, [110](#)
  - nppiMulCScale\_16u\_C4R, [110](#)
  - nppiMulCScale\_8u\_AC4IR, [111](#)
  - nppiMulCScale\_8u\_AC4R, [111](#)
  - nppiMulCScale\_8u\_C1IR, [111](#)
  - nppiMulCScale\_8u\_C1R, [112](#)

- [nppiMulCScale\\_8u\\_C3IR](#), [112](#)
  - [nppiMulCScale\\_8u\\_C3R](#), [112](#)
  - [nppiMulCScale\\_8u\\_C4IR](#), [113](#)
  - [nppiMulCScale\\_8u\\_C4R](#), [113](#)
- [image\\_mulscale](#)
  - [nppiMulScale\\_16u\\_AC4IR](#), [238](#)
  - [nppiMulScale\\_16u\\_AC4R](#), [239](#)
  - [nppiMulScale\\_16u\\_C1IR](#), [239](#)
  - [nppiMulScale\\_16u\\_C1R](#), [239](#)
  - [nppiMulScale\\_16u\\_C3IR](#), [240](#)
  - [nppiMulScale\\_16u\\_C3R](#), [240](#)
  - [nppiMulScale\\_16u\\_C4IR](#), [241](#)
  - [nppiMulScale\\_16u\\_C4R](#), [241](#)
  - [nppiMulScale\\_8u\\_AC4IR](#), [241](#)
  - [nppiMulScale\\_8u\\_AC4R](#), [242](#)
  - [nppiMulScale\\_8u\\_C1IR](#), [242](#)
  - [nppiMulScale\\_8u\\_C1R](#), [243](#)
  - [nppiMulScale\\_8u\\_C3IR](#), [243](#)
  - [nppiMulScale\\_8u\\_C3R](#), [243](#)
  - [nppiMulScale\\_8u\\_C4IR](#), [244](#)
  - [nppiMulScale\\_8u\\_C4R](#), [244](#)
- [image\\_not](#)
  - [nppiNot\\_8u\\_AC4IR](#), [468](#)
  - [nppiNot\\_8u\\_AC4R](#), [469](#)
  - [nppiNot\\_8u\\_C1IR](#), [469](#)
  - [nppiNot\\_8u\\_C1R](#), [469](#)
  - [nppiNot\\_8u\\_C3IR](#), [469](#)
  - [nppiNot\\_8u\\_C3R](#), [470](#)
  - [nppiNot\\_8u\\_C4IR](#), [470](#)
  - [nppiNot\\_8u\\_C4R](#), [470](#)
- [image\\_or](#)
  - [nppiOr\\_16u\\_AC4IR](#), [446](#)
  - [nppiOr\\_16u\\_AC4R](#), [446](#)
  - [nppiOr\\_16u\\_C1IR](#), [446](#)
  - [nppiOr\\_16u\\_C1R](#), [447](#)
  - [nppiOr\\_16u\\_C3IR](#), [447](#)
  - [nppiOr\\_16u\\_C3R](#), [447](#)
  - [nppiOr\\_16u\\_C4IR](#), [448](#)
  - [nppiOr\\_16u\\_C4R](#), [448](#)
  - [nppiOr\\_32s\\_AC4IR](#), [449](#)
  - [nppiOr\\_32s\\_AC4R](#), [449](#)
  - [nppiOr\\_32s\\_C1IR](#), [449](#)
  - [nppiOr\\_32s\\_C1R](#), [450](#)
  - [nppiOr\\_32s\\_C3IR](#), [450](#)
  - [nppiOr\\_32s\\_C3R](#), [450](#)
  - [nppiOr\\_32s\\_C4IR](#), [451](#)
  - [nppiOr\\_32s\\_C4R](#), [451](#)
  - [nppiOr\\_8u\\_AC4IR](#), [452](#)
  - [nppiOr\\_8u\\_AC4R](#), [452](#)
  - [nppiOr\\_8u\\_C1IR](#), [452](#)
  - [nppiOr\\_8u\\_C1R](#), [453](#)
  - [nppiOr\\_8u\\_C3IR](#), [453](#)
  - [nppiOr\\_8u\\_C3R](#), [453](#)
  - [nppiOr\\_8u\\_C4IR](#), [454](#)
  - [nppiOr\\_8u\\_C4R](#), [454](#)
- [image\\_orc](#)
  - [nppiOrC\\_16u\\_AC4IR](#), [384](#)
  - [nppiOrC\\_16u\\_AC4R](#), [384](#)
  - [nppiOrC\\_16u\\_C1IR](#), [384](#)
  - [nppiOrC\\_16u\\_C1R](#), [385](#)
  - [nppiOrC\\_16u\\_C3IR](#), [385](#)
  - [nppiOrC\\_16u\\_C3R](#), [385](#)
  - [nppiOrC\\_16u\\_C4IR](#), [386](#)
  - [nppiOrC\\_16u\\_C4R](#), [386](#)
  - [nppiOrC\\_32s\\_AC4IR](#), [386](#)
  - [nppiOrC\\_32s\\_AC4R](#), [387](#)
  - [nppiOrC\\_32s\\_C1IR](#), [387](#)
  - [nppiOrC\\_32s\\_C1R](#), [387](#)
  - [nppiOrC\\_32s\\_C3IR](#), [388](#)
  - [nppiOrC\\_32s\\_C3R](#), [388](#)
  - [nppiOrC\\_32s\\_C4IR](#), [388](#)
  - [nppiOrC\\_32s\\_C4R](#), [389](#)
  - [nppiOrC\\_8u\\_AC4IR](#), [389](#)
  - [nppiOrC\\_8u\\_AC4R](#), [389](#)
  - [nppiOrC\\_8u\\_C1IR](#), [390](#)
  - [nppiOrC\\_8u\\_C1R](#), [390](#)
  - [nppiOrC\\_8u\\_C3IR](#), [390](#)
  - [nppiOrC\\_8u\\_C3R](#), [391](#)
  - [nppiOrC\\_8u\\_C4IR](#), [391](#)
  - [nppiOrC\\_8u\\_C4R](#), [391](#)
- [image\\_perspective\\_transforms](#)
  - [nppiGetPerspectiveBound](#), [1325](#)
  - [nppiGetPerspectiveQuad](#), [1325](#)
  - [nppiGetPerspectiveTransform](#), [1326](#)
  - [nppiWarpPerspective\\_16u\\_AC4R](#), [1326](#)
  - [nppiWarpPerspective\\_16u\\_C1R](#), [1327](#)
  - [nppiWarpPerspective\\_16u\\_C3R](#), [1327](#)
  - [nppiWarpPerspective\\_16u\\_C4R](#), [1328](#)
  - [nppiWarpPerspective\\_16u\\_P3R](#), [1328](#)
  - [nppiWarpPerspective\\_16u\\_P4R](#), [1329](#)
  - [nppiWarpPerspective\\_32f\\_AC4R](#), [1329](#)
  - [nppiWarpPerspective\\_32f\\_C1R](#), [1330](#)
  - [nppiWarpPerspective\\_32f\\_C3R](#), [1330](#)
  - [nppiWarpPerspective\\_32f\\_C4R](#), [1331](#)
  - [nppiWarpPerspective\\_32f\\_P3R](#), [1331](#)
  - [nppiWarpPerspective\\_32f\\_P4R](#), [1332](#)
  - [nppiWarpPerspective\\_32s\\_AC4R](#), [1332](#)
  - [nppiWarpPerspective\\_32s\\_C1R](#), [1333](#)
  - [nppiWarpPerspective\\_32s\\_C3R](#), [1333](#)
  - [nppiWarpPerspective\\_32s\\_C4R](#), [1334](#)
  - [nppiWarpPerspective\\_32s\\_P3R](#), [1334](#)
  - [nppiWarpPerspective\\_32s\\_P4R](#), [1334](#)
  - [nppiWarpPerspective\\_8u\\_AC4R](#), [1335](#)
  - [nppiWarpPerspective\\_8u\\_C1R](#), [1335](#)
  - [nppiWarpPerspective\\_8u\\_C3R](#), [1336](#)
  - [nppiWarpPerspective\\_8u\\_C4R](#), [1336](#)
  - [nppiWarpPerspective\\_8u\\_P3R](#), [1337](#)
  - [nppiWarpPerspective\\_8u\\_P4R](#), [1337](#)

- [nppiWarpPerspectiveBack\\_16u\\_AC4R](#), 1338
- [nppiWarpPerspectiveBack\\_16u\\_C1R](#), 1338
- [nppiWarpPerspectiveBack\\_16u\\_C3R](#), 1339
- [nppiWarpPerspectiveBack\\_16u\\_C4R](#), 1339
- [nppiWarpPerspectiveBack\\_16u\\_P3R](#), 1340
- [nppiWarpPerspectiveBack\\_16u\\_P4R](#), 1340
- [nppiWarpPerspectiveBack\\_32f\\_AC4R](#), 1341
- [nppiWarpPerspectiveBack\\_32f\\_C1R](#), 1341
- [nppiWarpPerspectiveBack\\_32f\\_C3R](#), 1342
- [nppiWarpPerspectiveBack\\_32f\\_C4R](#), 1342
- [nppiWarpPerspectiveBack\\_32f\\_P3R](#), 1343
- [nppiWarpPerspectiveBack\\_32f\\_P4R](#), 1343
- [nppiWarpPerspectiveBack\\_32s\\_AC4R](#), 1344
- [nppiWarpPerspectiveBack\\_32s\\_C1R](#), 1344
- [nppiWarpPerspectiveBack\\_32s\\_C3R](#), 1345
- [nppiWarpPerspectiveBack\\_32s\\_C4R](#), 1345
- [nppiWarpPerspectiveBack\\_32s\\_P3R](#), 1346
- [nppiWarpPerspectiveBack\\_32s\\_P4R](#), 1346
- [nppiWarpPerspectiveBack\\_8u\\_AC4R](#), 1347
- [nppiWarpPerspectiveBack\\_8u\\_C1R](#), 1347
- [nppiWarpPerspectiveBack\\_8u\\_C3R](#), 1348
- [nppiWarpPerspectiveBack\\_8u\\_C4R](#), 1348
- [nppiWarpPerspectiveBack\\_8u\\_P3R](#), 1349
- [nppiWarpPerspectiveBack\\_8u\\_P4R](#), 1349
- [nppiWarpPerspectiveQuad\\_16u\\_AC4R](#), 1350
- [nppiWarpPerspectiveQuad\\_16u\\_C1R](#), 1350
- [nppiWarpPerspectiveQuad\\_16u\\_C3R](#), 1351
- [nppiWarpPerspectiveQuad\\_16u\\_C4R](#), 1351
- [nppiWarpPerspectiveQuad\\_16u\\_P3R](#), 1352
- [nppiWarpPerspectiveQuad\\_16u\\_P4R](#), 1352
- [nppiWarpPerspectiveQuad\\_32f\\_AC4R](#), 1353
- [nppiWarpPerspectiveQuad\\_32f\\_C1R](#), 1353
- [nppiWarpPerspectiveQuad\\_32f\\_C3R](#), 1354
- [nppiWarpPerspectiveQuad\\_32f\\_C4R](#), 1354
- [nppiWarpPerspectiveQuad\\_32f\\_P3R](#), 1355
- [nppiWarpPerspectiveQuad\\_32f\\_P4R](#), 1355
- [nppiWarpPerspectiveQuad\\_32s\\_AC4R](#), 1356
- [nppiWarpPerspectiveQuad\\_32s\\_C1R](#), 1356
- [nppiWarpPerspectiveQuad\\_32s\\_C3R](#), 1357
- [nppiWarpPerspectiveQuad\\_32s\\_C4R](#), 1357
- [nppiWarpPerspectiveQuad\\_32s\\_P3R](#), 1358
- [nppiWarpPerspectiveQuad\\_32s\\_P4R](#), 1358
- [nppiWarpPerspectiveQuad\\_8u\\_AC4R](#), 1359
- [nppiWarpPerspectiveQuad\\_8u\\_C1R](#), 1359
- [nppiWarpPerspectiveQuad\\_8u\\_C3R](#), 1360
- [nppiWarpPerspectiveQuad\\_8u\\_C4R](#), 1360
- [nppiWarpPerspectiveQuad\\_8u\\_P3R](#), 1361
- [nppiWarpPerspectiveQuad\\_8u\\_P4R](#), 1361
- [image\\_quality\\_index](#)
  - [nppiQualityIndex\\_16u32f\\_AC4R](#), 2017
  - [nppiQualityIndex\\_16u32f\\_C1R](#), 2017
  - [nppiQualityIndex\\_16u32f\\_C3R](#), 2018
  - [nppiQualityIndex\\_32f\\_AC4R](#), 2018
  - [nppiQualityIndex\\_32f\\_C1R](#), 2019
  - [nppiQualityIndex\\_32f\\_C3R](#), 2019
  - [nppiQualityIndex\\_8u32f\\_AC4R](#), 2019
  - [nppiQualityIndex\\_8u32f\\_C1R](#), 2020
  - [nppiQualityIndex\\_8u32f\\_C3R](#), 2020
  - [nppiQualityIndexGetBufferHostSize\\_-16u32f\\_AC4R](#), 2021
  - [nppiQualityIndexGetBufferHostSize\\_-16u32f\\_C1R](#), 2021
  - [nppiQualityIndexGetBufferHostSize\\_-16u32f\\_C3R](#), 2021
  - [nppiQualityIndexGetBufferHostSize\\_32f\\_-AC4R](#), 2022
  - [nppiQualityIndexGetBufferHostSize\\_32f\\_-C1R](#), 2022
  - [nppiQualityIndexGetBufferHostSize\\_32f\\_-C3R](#), 2022
  - [nppiQualityIndexGetBufferHostSize\\_8u32f\\_-AC4R](#), 2023
  - [nppiQualityIndexGetBufferHostSize\\_8u32f\\_-C1R](#), 2023
  - [nppiQualityIndexGetBufferHostSize\\_8u32f\\_-C3R](#), 2023
- [image\\_quantization](#)
  - [nppiDCTFree](#), 706
  - [nppiDCTInitAlloc](#), 706
  - [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_-C1R](#), 706
  - [nppiDCTQuantFwd8x8LS\\_JPEG\\_8u16s\\_-C1R\\_NEW](#), 707
  - [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_C1R](#), 707
  - [nppiDCTQuantInv8x8LS\\_JPEG\\_16s8u\\_-C1R\\_NEW](#), 708
  - [NppiDCTState](#), 706
  - [nppiQuantFwdRawTableInit\\_JPEG\\_8u](#), 708
  - [nppiQuantFwdTableInit\\_JPEG\\_8u16u](#), 709
  - [nppiQuantInvTableInit\\_JPEG\\_8u16u](#), 709
- [image\\_rank\\_filters](#)
  - [nppiFilterMax\\_16s\\_AC4R](#), 1126
  - [nppiFilterMax\\_16s\\_C1R](#), 1127
  - [nppiFilterMax\\_16s\\_C3R](#), 1127
  - [nppiFilterMax\\_16s\\_C4R](#), 1127
  - [nppiFilterMax\\_16u\\_AC4R](#), 1128
  - [nppiFilterMax\\_16u\\_C1R](#), 1128
  - [nppiFilterMax\\_16u\\_C3R](#), 1128
  - [nppiFilterMax\\_16u\\_C4R](#), 1129
  - [nppiFilterMax\\_32f\\_AC4R](#), 1129
  - [nppiFilterMax\\_32f\\_C1R](#), 1130
  - [nppiFilterMax\\_32f\\_C3R](#), 1130
  - [nppiFilterMax\\_32f\\_C4R](#), 1130
  - [nppiFilterMax\\_8u\\_AC4R](#), 1131
  - [nppiFilterMax\\_8u\\_C1R](#), 1131
  - [nppiFilterMax\\_8u\\_C3R](#), 1132
  - [nppiFilterMax\\_8u\\_C4R](#), 1132

- [nppiFilterMedian\\_16s\\_AC4R](#), 1132
- [nppiFilterMedian\\_16s\\_C1R](#), 1133
- [nppiFilterMedian\\_16s\\_C3R](#), 1133
- [nppiFilterMedian\\_16s\\_C4R](#), 1134
- [nppiFilterMedian\\_16u\\_AC4R](#), 1134
- [nppiFilterMedian\\_16u\\_C1R](#), 1135
- [nppiFilterMedian\\_16u\\_C3R](#), 1135
- [nppiFilterMedian\\_16u\\_C4R](#), 1135
- [nppiFilterMedian\\_32f\\_AC4R](#), 1136
- [nppiFilterMedian\\_32f\\_C1R](#), 1136
- [nppiFilterMedian\\_32f\\_C3R](#), 1137
- [nppiFilterMedian\\_32f\\_C4R](#), 1137
- [nppiFilterMedian\\_8u\\_AC4R](#), 1138
- [nppiFilterMedian\\_8u\\_C1R](#), 1138
- [nppiFilterMedian\\_8u\\_C3R](#), 1138
- [nppiFilterMedian\\_8u\\_C4R](#), 1139
- [nppiFilterMedianGetBufferSize\\_16s\\_AC4R](#), 1139
- [nppiFilterMedianGetBufferSize\\_16s\\_C1R](#), 1140
- [nppiFilterMedianGetBufferSize\\_16s\\_C3R](#), 1140
- [nppiFilterMedianGetBufferSize\\_16s\\_C4R](#), 1140
- [nppiFilterMedianGetBufferSize\\_16u\\_AC4R](#), 1140
- [nppiFilterMedianGetBufferSize\\_16u\\_C1R](#), 1141
- [nppiFilterMedianGetBufferSize\\_16u\\_C3R](#), 1141
- [nppiFilterMedianGetBufferSize\\_16u\\_C4R](#), 1141
- [nppiFilterMedianGetBufferSize\\_32f\\_AC4R](#), 1142
- [nppiFilterMedianGetBufferSize\\_32f\\_C1R](#), 1142
- [nppiFilterMedianGetBufferSize\\_32f\\_C3R](#), 1142
- [nppiFilterMedianGetBufferSize\\_32f\\_C4R](#), 1142
- [nppiFilterMedianGetBufferSize\\_8u\\_AC4R](#), 1143
- [nppiFilterMedianGetBufferSize\\_8u\\_C1R](#), 1143
- [nppiFilterMedianGetBufferSize\\_8u\\_C3R](#), 1143
- [nppiFilterMedianGetBufferSize\\_8u\\_C4R](#), 1144
- [nppiFilterMin\\_16s\\_AC4R](#), 1144
- [nppiFilterMin\\_16s\\_C1R](#), 1144
- [nppiFilterMin\\_16s\\_C3R](#), 1145
- [nppiFilterMin\\_16s\\_C4R](#), 1145
- [nppiFilterMin\\_16u\\_AC4R](#), 1146
- [nppiFilterMin\\_16u\\_C1R](#), 1146
- [nppiFilterMin\\_16u\\_C3R](#), 1146
- [nppiFilterMin\\_16u\\_C4R](#), 1147
- [nppiFilterMin\\_32f\\_AC4R](#), 1147
- [nppiFilterMin\\_32f\\_C1R](#), 1148
- [nppiFilterMin\\_32f\\_C3R](#), 1148
- [nppiFilterMin\\_32f\\_C4R](#), 1148
- [nppiFilterMin\\_8u\\_AC4R](#), 1149
- [nppiFilterMin\\_8u\\_C1R](#), 1149
- [nppiFilterMin\\_8u\\_C3R](#), 1150
- [nppiFilterMin\\_8u\\_C4R](#), 1150
- [image\\_rectstddev](#)
  - [nppiRectStdDev\\_32f\\_C1R](#), 1851
  - [nppiRectStdDev\\_32s32f\\_C1R](#), 1852
  - [nppiRectStdDev\\_32s\\_C1RSfs](#), 1852
- [image\\_remap](#)
  - [nppiRemap\\_16s\\_AC4R](#), 1222
  - [nppiRemap\\_16s\\_C1R](#), 1223
  - [nppiRemap\\_16s\\_C3R](#), 1223
  - [nppiRemap\\_16s\\_C4R](#), 1224
  - [nppiRemap\\_16s\\_P3R](#), 1225
  - [nppiRemap\\_16s\\_P4R](#), 1225
  - [nppiRemap\\_16u\\_AC4R](#), 1226
  - [nppiRemap\\_16u\\_C1R](#), 1226
  - [nppiRemap\\_16u\\_C3R](#), 1227
  - [nppiRemap\\_16u\\_C4R](#), 1228
  - [nppiRemap\\_16u\\_P3R](#), 1228
  - [nppiRemap\\_16u\\_P4R](#), 1229
  - [nppiRemap\\_32f\\_AC4R](#), 1229
  - [nppiRemap\\_32f\\_C1R](#), 1230
  - [nppiRemap\\_32f\\_C3R](#), 1231
  - [nppiRemap\\_32f\\_C4R](#), 1231
  - [nppiRemap\\_32f\\_P3R](#), 1232
  - [nppiRemap\\_32f\\_P4R](#), 1232
  - [nppiRemap\\_64f\\_AC4R](#), 1233
  - [nppiRemap\\_64f\\_C1R](#), 1234
  - [nppiRemap\\_64f\\_C3R](#), 1234
  - [nppiRemap\\_64f\\_C4R](#), 1235
  - [nppiRemap\\_64f\\_P3R](#), 1235
  - [nppiRemap\\_64f\\_P4R](#), 1236
  - [nppiRemap\\_8u\\_AC4R](#), 1237
  - [nppiRemap\\_8u\\_C1R](#), 1237
  - [nppiRemap\\_8u\\_C3R](#), 1238
  - [nppiRemap\\_8u\\_C4R](#), 1238
  - [nppiRemap\\_8u\\_P3R](#), 1239
  - [nppiRemap\\_8u\\_P4R](#), 1240
- [image\\_resize](#)
  - [nppiResize\\_16u\\_AC4R](#), 1209
  - [nppiResize\\_16u\\_C1R](#), 1210
  - [nppiResize\\_16u\\_C3R](#), 1210
  - [nppiResize\\_16u\\_C4R](#), 1211
  - [nppiResize\\_16u\\_P3R](#), 1211
  - [nppiResize\\_16u\\_P4R](#), 1212
  - [nppiResize\\_32f\\_AC4R](#), 1212
  - [nppiResize\\_32f\\_C1R](#), 1213



- nppiResize\_32f\_C3R, [1213](#)
- nppiResize\_32f\_C4R, [1214](#)
- nppiResize\_32f\_P3R, [1214](#)
- nppiResize\_32f\_P4R, [1215](#)
- nppiResize\_8u\_AC4R, [1215](#)
- nppiResize\_8u\_C1R, [1216](#)
- nppiResize\_8u\_C3R, [1216](#)
- nppiResize\_8u\_C4R, [1217](#)
- nppiResize\_8u\_P3R, [1217](#)
- nppiResize\_8u\_P4R, [1218](#)
- image\_resize\_square\_pixel
  - nppiGetResizeRect, [1189](#)
  - nppiResizeSqrPixel\_16s\_AC4R, [1189](#)
  - nppiResizeSqrPixel\_16s\_C1R, [1190](#)
  - nppiResizeSqrPixel\_16s\_C3R, [1190](#)
  - nppiResizeSqrPixel\_16s\_C4R, [1191](#)
  - nppiResizeSqrPixel\_16s\_P3R, [1191](#)
  - nppiResizeSqrPixel\_16s\_P4R, [1192](#)
  - nppiResizeSqrPixel\_16u\_AC4R, [1192](#)
  - nppiResizeSqrPixel\_16u\_C1R, [1193](#)
  - nppiResizeSqrPixel\_16u\_C3R, [1193](#)
  - nppiResizeSqrPixel\_16u\_C4R, [1194](#)
  - nppiResizeSqrPixel\_16u\_P3R, [1194](#)
  - nppiResizeSqrPixel\_16u\_P4R, [1195](#)
  - nppiResizeSqrPixel\_32f\_AC4R, [1196](#)
  - nppiResizeSqrPixel\_32f\_C1R, [1196](#)
  - nppiResizeSqrPixel\_32f\_C3R, [1197](#)
  - nppiResizeSqrPixel\_32f\_C4R, [1197](#)
  - nppiResizeSqrPixel\_32f\_P3R, [1198](#)
  - nppiResizeSqrPixel\_32f\_P4R, [1198](#)
  - nppiResizeSqrPixel\_64f\_AC4R, [1199](#)
  - nppiResizeSqrPixel\_64f\_C1R, [1200](#)
  - nppiResizeSqrPixel\_64f\_C3R, [1200](#)
  - nppiResizeSqrPixel\_64f\_C4R, [1201](#)
  - nppiResizeSqrPixel\_64f\_P3R, [1201](#)
  - nppiResizeSqrPixel\_64f\_P4R, [1202](#)
  - nppiResizeSqrPixel\_8u\_AC4R, [1202](#)
  - nppiResizeSqrPixel\_8u\_C1R, [1203](#)
  - nppiResizeSqrPixel\_8u\_C3R, [1203](#)
  - nppiResizeSqrPixel\_8u\_C4R, [1204](#)
  - nppiResizeSqrPixel\_8u\_P3R, [1204](#)
  - nppiResizeSqrPixel\_8u\_P4R, [1205](#)
- image\_rotate
  - nppiGetRotateBound, [1242](#)
  - nppiGetRotateQuad, [1243](#)
  - nppiRotate\_16u\_AC4R, [1243](#)
  - nppiRotate\_16u\_C1R, [1244](#)
  - nppiRotate\_16u\_C3R, [1244](#)
  - nppiRotate\_16u\_C4R, [1245](#)
  - nppiRotate\_32f\_AC4R, [1245](#)
  - nppiRotate\_32f\_C1R, [1246](#)
  - nppiRotate\_32f\_C3R, [1246](#)
  - nppiRotate\_32f\_C4R, [1247](#)
  - nppiRotate\_8u\_AC4R, [1247](#)
  - nppiRotate\_8u\_C1R, [1248](#)
  - nppiRotate\_8u\_C3R, [1248](#)
  - nppiRotate\_8u\_C4R, [1249](#)
- image\_rshifc
  - nppiRShiftC\_16s\_AC4IR, [407](#)
  - nppiRShiftC\_16s\_AC4R, [407](#)
  - nppiRShiftC\_16s\_C1IR, [408](#)
  - nppiRShiftC\_16s\_C1R, [408](#)
  - nppiRShiftC\_16s\_C3IR, [408](#)
  - nppiRShiftC\_16s\_C3R, [409](#)
  - nppiRShiftC\_16s\_C4IR, [409](#)
  - nppiRShiftC\_16s\_C4R, [409](#)
  - nppiRShiftC\_16u\_AC4IR, [410](#)
  - nppiRShiftC\_16u\_AC4R, [410](#)
  - nppiRShiftC\_16u\_C1IR, [410](#)
  - nppiRShiftC\_16u\_C1R, [411](#)
  - nppiRShiftC\_16u\_C3IR, [411](#)
  - nppiRShiftC\_16u\_C3R, [411](#)
  - nppiRShiftC\_16u\_C4IR, [412](#)
  - nppiRShiftC\_16u\_C4R, [412](#)
  - nppiRShiftC\_32s\_AC4IR, [412](#)
  - nppiRShiftC\_32s\_AC4R, [413](#)
  - nppiRShiftC\_32s\_C1IR, [413](#)
  - nppiRShiftC\_32s\_C1R, [413](#)
  - nppiRShiftC\_32s\_C3IR, [414](#)
  - nppiRShiftC\_32s\_C3R, [414](#)
  - nppiRShiftC\_32s\_C4IR, [414](#)
  - nppiRShiftC\_32s\_C4R, [415](#)
  - nppiRShiftC\_8s\_AC4IR, [415](#)
  - nppiRShiftC\_8s\_AC4R, [415](#)
  - nppiRShiftC\_8s\_C1IR, [416](#)
  - nppiRShiftC\_8s\_C1R, [416](#)
  - nppiRShiftC\_8s\_C3IR, [416](#)
  - nppiRShiftC\_8s\_C3R, [417](#)
  - nppiRShiftC\_8s\_C4IR, [417](#)
  - nppiRShiftC\_8s\_C4R, [417](#)
  - nppiRShiftC\_8u\_AC4IR, [418](#)
  - nppiRShiftC\_8u\_AC4R, [418](#)
  - nppiRShiftC\_8u\_C1IR, [418](#)
  - nppiRShiftC\_8u\_C1R, [419](#)
  - nppiRShiftC\_8u\_C3IR, [419](#)
  - nppiRShiftC\_8u\_C3R, [419](#)
  - nppiRShiftC\_8u\_C4IR, [420](#)
  - nppiRShiftC\_8u\_C4R, [420](#)
- image\_scale
  - nppiScale\_16s8u\_AC4R, [848](#)
  - nppiScale\_16s8u\_C1R, [848](#)
  - nppiScale\_16s8u\_C3R, [848](#)
  - nppiScale\_16s8u\_C4R, [849](#)
  - nppiScale\_16u8u\_AC4R, [849](#)
  - nppiScale\_16u8u\_C1R, [849](#)
  - nppiScale\_16u8u\_C3R, [850](#)
  - nppiScale\_16u8u\_C4R, [850](#)
  - nppiScale\_32f8u\_AC4R, [850](#)

- nppiScale\_32f8u\_C1R, [851](#)
- nppiScale\_32f8u\_C3R, [851](#)
- nppiScale\_32f8u\_C4R, [852](#)
- nppiScale\_32s8u\_AC4R, [852](#)
- nppiScale\_32s8u\_C1R, [852](#)
- nppiScale\_32s8u\_C3R, [853](#)
- nppiScale\_32s8u\_C4R, [853](#)
- nppiScale\_8u16s\_AC4R, [853](#)
- nppiScale\_8u16s\_C1R, [854](#)
- nppiScale\_8u16s\_C3R, [854](#)
- nppiScale\_8u16s\_C4R, [854](#)
- nppiScale\_8u16u\_AC4R, [855](#)
- nppiScale\_8u16u\_C1R, [855](#)
- nppiScale\_8u16u\_C3R, [855](#)
- nppiScale\_8u16u\_C4R, [856](#)
- nppiScale\_8u32f\_AC4R, [856](#)
- nppiScale\_8u32f\_C1R, [856](#)
- nppiScale\_8u32f\_C3R, [857](#)
- nppiScale\_8u32f\_C4R, [857](#)
- nppiScale\_8u32s\_AC4R, [858](#)
- nppiScale\_8u32s\_C1R, [858](#)
- nppiScale\_8u32s\_C3R, [858](#)
- nppiScale\_8u32s\_C4R, [859](#)
- image\_set
  - nppiSet\_16s\_AC4MR, [726](#)
  - nppiSet\_16s\_AC4R, [727](#)
  - nppiSet\_16s\_C1MR, [727](#)
  - nppiSet\_16s\_C1R, [727](#)
  - nppiSet\_16s\_C2R, [728](#)
  - nppiSet\_16s\_C3CR, [728](#)
  - nppiSet\_16s\_C3MR, [728](#)
  - nppiSet\_16s\_C3R, [729](#)
  - nppiSet\_16s\_C4CR, [729](#)
  - nppiSet\_16s\_C4MR, [729](#)
  - nppiSet\_16s\_C4R, [730](#)
  - nppiSet\_16sc\_AC4R, [730](#)
  - nppiSet\_16sc\_C1R, [730](#)
  - nppiSet\_16sc\_C2R, [731](#)
  - nppiSet\_16sc\_C3R, [731](#)
  - nppiSet\_16sc\_C4R, [731](#)
  - nppiSet\_16u\_AC4MR, [732](#)
  - nppiSet\_16u\_AC4R, [732](#)
  - nppiSet\_16u\_C1MR, [732](#)
  - nppiSet\_16u\_C1R, [733](#)
  - nppiSet\_16u\_C2R, [733](#)
  - nppiSet\_16u\_C3CR, [733](#)
  - nppiSet\_16u\_C3MR, [734](#)
  - nppiSet\_16u\_C3R, [734](#)
  - nppiSet\_16u\_C4CR, [734](#)
  - nppiSet\_16u\_C4MR, [735](#)
  - nppiSet\_16u\_C4R, [735](#)
  - nppiSet\_32f\_AC4MR, [735](#)
  - nppiSet\_32f\_AC4R, [736](#)
  - nppiSet\_32f\_C1MR, [736](#)
  - nppiSet\_32f\_C1R, [736](#)
  - nppiSet\_32f\_C2R, [737](#)
  - nppiSet\_32f\_C3CR, [737](#)
  - nppiSet\_32f\_C3MR, [737](#)
  - nppiSet\_32f\_C3R, [738](#)
  - nppiSet\_32f\_C4CR, [738](#)
  - nppiSet\_32f\_C4MR, [738](#)
  - nppiSet\_32f\_C4R, [739](#)
  - nppiSet\_32fc\_AC4R, [739](#)
  - nppiSet\_32fc\_C1R, [739](#)
  - nppiSet\_32fc\_C2R, [740](#)
  - nppiSet\_32fc\_C3R, [740](#)
  - nppiSet\_32fc\_C4R, [740](#)
  - nppiSet\_32s\_AC4MR, [741](#)
  - nppiSet\_32s\_AC4R, [741](#)
  - nppiSet\_32s\_C1MR, [741](#)
  - nppiSet\_32s\_C1R, [742](#)
  - nppiSet\_32s\_C2R, [742](#)
  - nppiSet\_32s\_C3CR, [742](#)
  - nppiSet\_32s\_C3MR, [743](#)
  - nppiSet\_32s\_C3R, [743](#)
  - nppiSet\_32s\_C4CR, [743](#)
  - nppiSet\_32s\_C4MR, [744](#)
  - nppiSet\_32s\_C4R, [744](#)
  - nppiSet\_32sc\_AC4R, [744](#)
  - nppiSet\_32sc\_C1R, [745](#)
  - nppiSet\_32sc\_C2R, [745](#)
  - nppiSet\_32sc\_C3R, [745](#)
  - nppiSet\_32sc\_C4R, [746](#)
  - nppiSet\_32u\_AC4R, [746](#)
  - nppiSet\_32u\_C1R, [746](#)
  - nppiSet\_32u\_C2R, [747](#)
  - nppiSet\_32u\_C3R, [747](#)
  - nppiSet\_32u\_C4R, [747](#)
  - nppiSet\_8s\_AC4R, [748](#)
  - nppiSet\_8s\_C1R, [748](#)
  - nppiSet\_8s\_C2R, [748](#)
  - nppiSet\_8s\_C3R, [749](#)
  - nppiSet\_8s\_C4R, [749](#)
  - nppiSet\_8u\_AC4MR, [749](#)
  - nppiSet\_8u\_AC4R, [750](#)
  - nppiSet\_8u\_C1MR, [750](#)
  - nppiSet\_8u\_C1R, [750](#)
  - nppiSet\_8u\_C2R, [751](#)
  - nppiSet\_8u\_C3CR, [751](#)
  - nppiSet\_8u\_C3MR, [751](#)
  - nppiSet\_8u\_C3R, [752](#)
  - nppiSet\_8u\_C4CR, [752](#)
  - nppiSet\_8u\_C4MR, [752](#)
  - nppiSet\_8u\_C4R, [753](#)
- image\_sqr
  - nppiSqr\_16s\_AC4IRSfs, [333](#)
  - nppiSqr\_16s\_AC4RSfs, [333](#)
  - nppiSqr\_16s\_C1IRSfs, [333](#)

- nppiSqr\_16s\_C1RSfs, [333](#)
- nppiSqr\_16s\_C3RSfs, [334](#)
- nppiSqr\_16s\_C3RSfs, [334](#)
- nppiSqr\_16s\_C4IRSfs, [334](#)
- nppiSqr\_16s\_C4RSfs, [335](#)
- nppiSqr\_16u\_AC4IRSfs, [335](#)
- nppiSqr\_16u\_AC4RSfs, [335](#)
- nppiSqr\_16u\_C1IRSfs, [336](#)
- nppiSqr\_16u\_C1RSfs, [336](#)
- nppiSqr\_16u\_C3IRSfs, [337](#)
- nppiSqr\_16u\_C3RSfs, [337](#)
- nppiSqr\_16u\_C4IRSfs, [337](#)
- nppiSqr\_16u\_C4RSfs, [338](#)
- nppiSqr\_32f\_AC4IR, [338](#)
- nppiSqr\_32f\_AC4R, [338](#)
- nppiSqr\_32f\_C1IR, [339](#)
- nppiSqr\_32f\_C1R, [339](#)
- nppiSqr\_32f\_C3IR, [339](#)
- nppiSqr\_32f\_C3R, [339](#)
- nppiSqr\_32f\_C4IR, [340](#)
- nppiSqr\_32f\_C4R, [340](#)
- nppiSqr\_8u\_AC4IRSfs, [340](#)
- nppiSqr\_8u\_AC4RSfs, [341](#)
- nppiSqr\_8u\_C1IRSfs, [341](#)
- nppiSqr\_8u\_C1RSfs, [341](#)
- nppiSqr\_8u\_C3IRSfs, [342](#)
- nppiSqr\_8u\_C3RSfs, [342](#)
- nppiSqr\_8u\_C4IRSfs, [342](#)
- nppiSqr\_8u\_C4RSfs, [343](#)
- image\_sqrintegral
  - nppiSqrIntegral\_8u32f64f\_C1R, [1848](#)
  - nppiSqrIntegral\_8u32s64f\_C1R, [1849](#)
  - nppiSqrIntegral\_8u32s\_C1R, [1849](#)
- image\_sqrt
  - nppiSqrt\_16s\_AC4IRSfs, [346](#)
  - nppiSqrt\_16s\_AC4RSfs, [346](#)
  - nppiSqrt\_16s\_C1IRSfs, [347](#)
  - nppiSqrt\_16s\_C1RSfs, [347](#)
  - nppiSqrt\_16s\_C3IRSfs, [348](#)
  - nppiSqrt\_16s\_C3RSfs, [348](#)
  - nppiSqrt\_16u\_AC4IRSfs, [348](#)
  - nppiSqrt\_16u\_AC4RSfs, [349](#)
  - nppiSqrt\_16u\_C1IRSfs, [349](#)
  - nppiSqrt\_16u\_C1RSfs, [349](#)
  - nppiSqrt\_16u\_C3IRSfs, [350](#)
  - nppiSqrt\_16u\_C3RSfs, [350](#)
  - nppiSqrt\_32f\_AC4IR, [350](#)
  - nppiSqrt\_32f\_AC4R, [351](#)
  - nppiSqrt\_32f\_C1IR, [351](#)
  - nppiSqrt\_32f\_C1R, [351](#)
  - nppiSqrt\_32f\_C3IR, [352](#)
  - nppiSqrt\_32f\_C3R, [352](#)
  - nppiSqrt\_32f\_C4IR, [352](#)
  - nppiSqrt\_32f\_C4R, [353](#)
  - nppiSqrt\_8u\_AC4IRSfs, [353](#)
  - nppiSqrt\_8u\_AC4RSfs, [353](#)
  - nppiSqrt\_8u\_C1IRSfs, [354](#)
  - nppiSqrt\_8u\_C1RSfs, [354](#)
  - nppiSqrt\_8u\_C3IRSfs, [355](#)
  - nppiSqrt\_8u\_C3RSfs, [355](#)
- image\_statistics\_functions
  - nppiAverageErrorGetBufferHostSize\_16s\_-C1R, [1409](#)
  - nppiAverageErrorGetBufferHostSize\_16s\_-C2R, [1409](#)
  - nppiAverageErrorGetBufferHostSize\_16s\_-C3R, [1409](#)
  - nppiAverageErrorGetBufferHostSize\_16s\_-C4R, [1409](#)
  - nppiAverageErrorGetBufferHostSize\_16sc\_-C1R, [1410](#)
  - nppiAverageErrorGetBufferHostSize\_16sc\_-C2R, [1410](#)
  - nppiAverageErrorGetBufferHostSize\_16sc\_-C3R, [1410](#)
  - nppiAverageErrorGetBufferHostSize\_16sc\_-C4R, [1411](#)
  - nppiAverageErrorGetBufferHostSize\_16u\_-C1R, [1411](#)
  - nppiAverageErrorGetBufferHostSize\_16u\_-C2R, [1411](#)
  - nppiAverageErrorGetBufferHostSize\_16u\_-C3R, [1411](#)
  - nppiAverageErrorGetBufferHostSize\_16u\_-C4R, [1412](#)
  - nppiAverageErrorGetBufferHostSize\_32f\_-C1R, [1412](#)
  - nppiAverageErrorGetBufferHostSize\_32f\_-C2R, [1412](#)
  - nppiAverageErrorGetBufferHostSize\_32f\_-C3R, [1413](#)
  - nppiAverageErrorGetBufferHostSize\_32f\_-C4R, [1413](#)
  - nppiAverageErrorGetBufferHostSize\_32fc\_-C1R, [1413](#)
  - nppiAverageErrorGetBufferHostSize\_32fc\_-C2R, [1413](#)
  - nppiAverageErrorGetBufferHostSize\_32fc\_-C3R, [1414](#)
  - nppiAverageErrorGetBufferHostSize\_32fc\_-C4R, [1414](#)
  - nppiAverageErrorGetBufferHostSize\_32s\_-C1R, [1414](#)
  - nppiAverageErrorGetBufferHostSize\_32s\_-C2R, [1415](#)
  - nppiAverageErrorGetBufferHostSize\_32s\_-C3R, [1415](#)



- [nppiAverageErrorGetBufferHostSize\\_32s\\_-C4R, 1415](#)
- [nppiAverageErrorGetBufferHostSize\\_32sc\\_-C1R, 1415](#)
- [nppiAverageErrorGetBufferHostSize\\_32sc\\_-C2R, 1416](#)
- [nppiAverageErrorGetBufferHostSize\\_32sc\\_-C3R, 1416](#)
- [nppiAverageErrorGetBufferHostSize\\_32sc\\_-C4R, 1416](#)
- [nppiAverageErrorGetBufferHostSize\\_32u\\_-C1R, 1417](#)
- [nppiAverageErrorGetBufferHostSize\\_32u\\_-C2R, 1417](#)
- [nppiAverageErrorGetBufferHostSize\\_32u\\_-C3R, 1417](#)
- [nppiAverageErrorGetBufferHostSize\\_32u\\_-C4R, 1417](#)
- [nppiAverageErrorGetBufferHostSize\\_64f\\_-C1R, 1418](#)
- [nppiAverageErrorGetBufferHostSize\\_64f\\_-C2R, 1418](#)
- [nppiAverageErrorGetBufferHostSize\\_64f\\_-C3R, 1418](#)
- [nppiAverageErrorGetBufferHostSize\\_64f\\_-C4R, 1419](#)
- [nppiAverageErrorGetBufferHostSize\\_8s\\_-C1R, 1419](#)
- [nppiAverageErrorGetBufferHostSize\\_8s\\_-C2R, 1419](#)
- [nppiAverageErrorGetBufferHostSize\\_8s\\_-C3R, 1419](#)
- [nppiAverageErrorGetBufferHostSize\\_8s\\_-C4R, 1420](#)
- [nppiAverageErrorGetBufferHostSize\\_8u\\_-C1R, 1420](#)
- [nppiAverageErrorGetBufferHostSize\\_8u\\_-C2R, 1420](#)
- [nppiAverageErrorGetBufferHostSize\\_8u\\_-C3R, 1421](#)
- [nppiAverageErrorGetBufferHostSize\\_8u\\_-C4R, 1421](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16s\\_C1R, 1421](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16s\\_C2R, 1421](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16s\\_C3R, 1422](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16s\\_C4R, 1422](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16sc\\_C1R, 1422](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16sc\\_C2R, 1423](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16sc\\_C3R, 1423](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16sc\\_C4R, 1423](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16u\\_C1R, 1423](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16u\\_C2R, 1424](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16u\\_C3R, 1424](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-16u\\_C4R, 1424](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32f\\_C1R, 1425](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32f\\_C2R, 1425](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32f\\_C3R, 1425](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32f\\_C4R, 1425](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32fc\\_C1R, 1426](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32fc\\_C2R, 1426](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32fc\\_C3R, 1426](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32fc\\_C4R, 1427](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32s\\_C1R, 1427](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32s\\_C2R, 1427](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32s\\_C3R, 1427](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32s\\_C4R, 1428](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32sc\\_C1R, 1428](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32sc\\_C2R, 1428](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32sc\\_C3R, 1429](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32sc\\_C4R, 1429](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32u\\_C1R, 1429](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32u\\_C2R, 1429](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32u\\_C3R, 1430](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-32u\\_C4R, 1430](#)
- [nppiAverageRelativeErrorGetBufferHostSize\\_-64f\\_C1R, 1430](#)

- [nppiAverageRelativeErrorGetBufferHostSize\\_64f\\_C2R](#), 1431
- [nppiAverageRelativeErrorGetBufferHostSize\\_64f\\_C3R](#), 1431
- [nppiAverageRelativeErrorGetBufferHostSize\\_64f\\_C4R](#), 1431
- [nppiAverageRelativeErrorGetBufferHostSize\\_8s\\_C1R](#), 1431
- [nppiAverageRelativeErrorGetBufferHostSize\\_8s\\_C2R](#), 1432
- [nppiAverageRelativeErrorGetBufferHostSize\\_8s\\_C3R](#), 1432
- [nppiAverageRelativeErrorGetBufferHostSize\\_8s\\_C4R](#), 1432
- [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C1R](#), 1433
- [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C2R](#), 1433
- [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C3R](#), 1433
- [nppiAverageRelativeErrorGetBufferHostSize\\_8u\\_C4R](#), 1433
- [nppiMaximumErrorGetBufferHostSize\\_16s\\_C1R](#), 1434
- [nppiMaximumErrorGetBufferHostSize\\_16s\\_C2R](#), 1434
- [nppiMaximumErrorGetBufferHostSize\\_16s\\_C3R](#), 1434
- [nppiMaximumErrorGetBufferHostSize\\_16s\\_C4R](#), 1435
- [nppiMaximumErrorGetBufferHostSize\\_16sc\\_C1R](#), 1435
- [nppiMaximumErrorGetBufferHostSize\\_16sc\\_C2R](#), 1435
- [nppiMaximumErrorGetBufferHostSize\\_16sc\\_C3R](#), 1435
- [nppiMaximumErrorGetBufferHostSize\\_16sc\\_C4R](#), 1436
- [nppiMaximumErrorGetBufferHostSize\\_16u\\_C1R](#), 1436
- [nppiMaximumErrorGetBufferHostSize\\_16u\\_C2R](#), 1436
- [nppiMaximumErrorGetBufferHostSize\\_16u\\_C3R](#), 1437
- [nppiMaximumErrorGetBufferHostSize\\_16u\\_C4R](#), 1437
- [nppiMaximumErrorGetBufferHostSize\\_32f\\_C1R](#), 1437
- [nppiMaximumErrorGetBufferHostSize\\_32f\\_C2R](#), 1437
- [nppiMaximumErrorGetBufferHostSize\\_32f\\_C3R](#), 1438
- [nppiMaximumErrorGetBufferHostSize\\_32f\\_C4R](#), 1438
- [nppiMaximumErrorGetBufferHostSize\\_32fc\\_C1R](#), 1438
- [nppiMaximumErrorGetBufferHostSize\\_32fc\\_C2R](#), 1439
- [nppiMaximumErrorGetBufferHostSize\\_32fc\\_C3R](#), 1439
- [nppiMaximumErrorGetBufferHostSize\\_32fc\\_C4R](#), 1439
- [nppiMaximumErrorGetBufferHostSize\\_32s\\_C1R](#), 1439
- [nppiMaximumErrorGetBufferHostSize\\_32s\\_C2R](#), 1440
- [nppiMaximumErrorGetBufferHostSize\\_32s\\_C3R](#), 1440
- [nppiMaximumErrorGetBufferHostSize\\_32s\\_C4R](#), 1440
- [nppiMaximumErrorGetBufferHostSize\\_32sc\\_C1R](#), 1441
- [nppiMaximumErrorGetBufferHostSize\\_32sc\\_C2R](#), 1441
- [nppiMaximumErrorGetBufferHostSize\\_32sc\\_C3R](#), 1441
- [nppiMaximumErrorGetBufferHostSize\\_32sc\\_C4R](#), 1441
- [nppiMaximumErrorGetBufferHostSize\\_32u\\_C1R](#), 1442
- [nppiMaximumErrorGetBufferHostSize\\_32u\\_C2R](#), 1442
- [nppiMaximumErrorGetBufferHostSize\\_32u\\_C3R](#), 1442
- [nppiMaximumErrorGetBufferHostSize\\_32u\\_C4R](#), 1443
- [nppiMaximumErrorGetBufferHostSize\\_64f\\_C1R](#), 1443
- [nppiMaximumErrorGetBufferHostSize\\_64f\\_C2R](#), 1443
- [nppiMaximumErrorGetBufferHostSize\\_64f\\_C3R](#), 1443
- [nppiMaximumErrorGetBufferHostSize\\_64f\\_C4R](#), 1444
- [nppiMaximumErrorGetBufferHostSize\\_8s\\_C1R](#), 1444
- [nppiMaximumErrorGetBufferHostSize\\_8s\\_C2R](#), 1444
- [nppiMaximumErrorGetBufferHostSize\\_8s\\_C3R](#), 1445
- [nppiMaximumErrorGetBufferHostSize\\_8s\\_C4R](#), 1445
- [nppiMaximumErrorGetBufferHostSize\\_8u\\_C1R](#), 1445
- [nppiMaximumErrorGetBufferHostSize\\_8u\\_C2R](#), 1445
- [nppiMaximumErrorGetBufferHostSize\\_8u\\_C3R](#), 1446

- [nppiMaximumErrorGetBufferHostSize\\_8u\\_-C4R, 1446](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16s\\_C1R, 1446](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16s\\_C2R, 1447](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16s\\_C3R, 1447](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16s\\_C4R, 1447](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16sc\\_C1R, 1447](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16sc\\_C2R, 1448](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16sc\\_C3R, 1448](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16sc\\_C4R, 1448](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16u\\_C1R, 1449](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16u\\_C2R, 1449](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16u\\_C3R, 1449](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-16u\\_C4R, 1449](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32f\\_C1R, 1450](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32f\\_C2R, 1450](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32f\\_C3R, 1450](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32f\\_C4R, 1451](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32fc\\_C1R, 1451](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-image\\_sub32fc\\_C2R, 1451](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32fc\\_C3R, 1451](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32fc\\_C4R, 1452](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32s\\_C1R, 1452](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32s\\_C2R, 1452](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32s\\_C3R, 1453](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32s\\_C4R, 1453](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32sc\\_C1R, 1453](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32sc\\_C2R, 1453](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32sc\\_C3R, 1454](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32sc\\_C4R, 1454](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32u\\_C1R, 1454](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32u\\_C2R, 1455](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32u\\_C3R, 1455](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-32u\\_C4R, 1455](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-64f\\_C1R, 1455](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-64f\\_C2R, 1456](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-64f\\_C3R, 1456](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-64f\\_C4R, 1456](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8s\\_C1R, 1457](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8s\\_C2R, 1457](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8s\\_C3R, 1457](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8s\\_C4R, 1457](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8u\\_C1R, 1458](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8u\\_C2R, 1458](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8u\\_C3R, 1458](#)
- [nppiMaximumRelativeErrorGetBufferHostSize\\_-8u\\_C4R, 1459](#)
- [nppiSub\\_16s\\_AC4IRSfs, 251](#)
- [nppiSub\\_16s\\_AC4RSfs, 252](#)
- [nppiSub\\_16s\\_C1IRSfs, 252](#)
- [nppiSub\\_16s\\_C1RSfs, 252](#)
- [nppiSub\\_16s\\_C3IRSfs, 253](#)
- [nppiSub\\_16s\\_C3RSfs, 253](#)
- [nppiSub\\_16s\\_C4IRSfs, 254](#)
- [nppiSub\\_16s\\_C4RSfs, 254](#)
- [nppiSub\\_16sc\\_AC4IRSfs, 254](#)
- [nppiSub\\_16sc\\_AC4RSfs, 255](#)
- [nppiSub\\_16sc\\_C1IRSfs, 255](#)
- [nppiSub\\_16sc\\_C1RSfs, 256](#)
- [nppiSub\\_16sc\\_C3IRSfs, 256](#)
- [nppiSub\\_16sc\\_C3RSfs, 256](#)
- [nppiSub\\_16u\\_AC4IRSfs, 257](#)
- [nppiSub\\_16u\\_AC4RSfs, 257](#)
- [nppiSub\\_16u\\_C1IRSfs, 258](#)

- nppiSub\_16u\_C1RSfs, 258
- nppiSub\_16u\_C3RSfs, 259
- nppiSub\_16u\_C3RSfs, 259
- nppiSub\_16u\_C4IRSfs, 259
- nppiSub\_16u\_C4RSfs, 260
- nppiSub\_32f\_AC4IR, 260
- nppiSub\_32f\_AC4R, 261
- nppiSub\_32f\_C1IR, 261
- nppiSub\_32f\_C1R, 261
- nppiSub\_32f\_C3IR, 262
- nppiSub\_32f\_C3R, 262
- nppiSub\_32f\_C4IR, 263
- nppiSub\_32f\_C4R, 263
- nppiSub\_32fc\_AC4IR, 263
- nppiSub\_32fc\_AC4R, 264
- nppiSub\_32fc\_C1IR, 264
- nppiSub\_32fc\_C1R, 265
- nppiSub\_32fc\_C3IR, 265
- nppiSub\_32fc\_C3R, 265
- nppiSub\_32fc\_C4IR, 266
- nppiSub\_32fc\_C4R, 266
- nppiSub\_32s\_C1IRSfs, 267
- nppiSub\_32s\_C1R, 267
- nppiSub\_32s\_C1RSfs, 267
- nppiSub\_32s\_C3IRSfs, 268
- nppiSub\_32s\_C3RSfs, 268
- nppiSub\_32s\_C4IRSfs, 269
- nppiSub\_32s\_C4RSfs, 269
- nppiSub\_32sc\_AC4IRSfs, 270
- nppiSub\_32sc\_AC4RSfs, 270
- nppiSub\_32sc\_C1IRSfs, 270
- nppiSub\_32sc\_C1RSfs, 271
- nppiSub\_32sc\_C3IRSfs, 271
- nppiSub\_32sc\_C3RSfs, 272
- nppiSub\_8u\_AC4IRSfs, 272
- nppiSub\_8u\_AC4RSfs, 272
- nppiSub\_8u\_C1IRSfs, 273
- nppiSub\_8u\_C1RSfs, 273
- nppiSub\_8u\_C3IRSfs, 274
- nppiSub\_8u\_C3RSfs, 274
- nppiSub\_8u\_C4IRSfs, 274
- nppiSub\_8u\_C4RSfs, 275
- image\_subc
  - nppiSubC\_16s\_AC4IRSfs, 119
  - nppiSubC\_16s\_AC4RSfs, 119
  - nppiSubC\_16s\_C1IRSfs, 119
  - nppiSubC\_16s\_C1RSfs, 120
  - nppiSubC\_16s\_C3IRSfs, 120
  - nppiSubC\_16s\_C3RSfs, 120
  - nppiSubC\_16s\_C4IRSfs, 121
  - nppiSubC\_16s\_C4RSfs, 121
  - nppiSubC\_16sc\_AC4IRSfs, 122
  - nppiSubC\_16sc\_AC4RSfs, 122
  - nppiSubC\_16sc\_C1IRSfs, 122
- nppiSubC\_16sc\_C1RSfs, 123
- nppiSubC\_16sc\_C3IRSfs, 123
- nppiSubC\_16sc\_C3RSfs, 124
- nppiSubC\_16u\_AC4IRSfs, 124
- nppiSubC\_16u\_AC4RSfs, 124
- nppiSubC\_16u\_C1IRSfs, 125
- nppiSubC\_16u\_C1RSfs, 125
- nppiSubC\_16u\_C3IRSfs, 126
- nppiSubC\_16u\_C3RSfs, 126
- nppiSubC\_16u\_C4IRSfs, 126
- nppiSubC\_16u\_C4RSfs, 127
- nppiSubC\_32f\_AC4IR, 127
- nppiSubC\_32f\_AC4R, 127
- nppiSubC\_32f\_C1IR, 128
- nppiSubC\_32f\_C1R, 128
- nppiSubC\_32f\_C3IR, 128
- nppiSubC\_32f\_C3R, 129
- nppiSubC\_32f\_C4IR, 129
- nppiSubC\_32f\_C4R, 129
- nppiSubC\_32fc\_AC4IR, 130
- nppiSubC\_32fc\_AC4R, 130
- nppiSubC\_32fc\_C1IR, 130
- nppiSubC\_32fc\_C1R, 131
- nppiSubC\_32fc\_C3IR, 131
- nppiSubC\_32fc\_C3R, 131
- nppiSubC\_32fc\_C4IR, 132
- nppiSubC\_32fc\_C4R, 132
- nppiSubC\_32s\_C1IRSfs, 133
- nppiSubC\_32s\_C1RSfs, 133
- nppiSubC\_32s\_C3IRSfs, 133
- nppiSubC\_32s\_C3RSfs, 134
- nppiSubC\_32sc\_AC4IRSfs, 134
- nppiSubC\_32sc\_AC4RSfs, 134
- nppiSubC\_32sc\_C1IRSfs, 135
- nppiSubC\_32sc\_C1RSfs, 135
- nppiSubC\_32sc\_C3IRSfs, 136
- nppiSubC\_32sc\_C3RSfs, 136
- nppiSubC\_8u\_AC4IRSfs, 136
- nppiSubC\_8u\_AC4RSfs, 137
- nppiSubC\_8u\_C1IRSfs, 137
- nppiSubC\_8u\_C1RSfs, 138
- nppiSubC\_8u\_C3IRSfs, 138
- nppiSubC\_8u\_C3RSfs, 138
- nppiSubC\_8u\_C4IRSfs, 139
- nppiSubC\_8u\_C4RSfs, 139
- image\_sum
  - nppiSum\_16s\_AC4R, 1463
  - nppiSum\_16s\_C1R, 1463
  - nppiSum\_16s\_C3R, 1463
  - nppiSum\_16s\_C4R, 1464
  - nppiSum\_16u\_AC4R, 1464
  - nppiSum\_16u\_C1R, 1464
  - nppiSum\_16u\_C3R, 1465
  - nppiSum\_16u\_C4R, 1465

- [nppiSum\\_32f\\_AC4R, 1465](#)
- [nppiSum\\_32f\\_C1R, 1466](#)
- [nppiSum\\_32f\\_C3R, 1466](#)
- [nppiSum\\_32f\\_C4R, 1466](#)
- [nppiSum\\_8u64s\\_C1R, 1467](#)
- [nppiSum\\_8u64s\\_C4R, 1467](#)
- [nppiSum\\_8u\\_AC4R, 1468](#)
- [nppiSum\\_8u\\_C1R, 1468](#)
- [nppiSum\\_8u\\_C3R, 1468](#)
- [nppiSum\\_8u\\_C4R, 1469](#)
- [nppiSumGetBufferHostSize\\_16s\\_AC4R, 1469](#)
- [nppiSumGetBufferHostSize\\_16s\\_C1R, 1469](#)
- [nppiSumGetBufferHostSize\\_16s\\_C3R, 1470](#)
- [nppiSumGetBufferHostSize\\_16s\\_C4R, 1470](#)
- [nppiSumGetBufferHostSize\\_16u\\_AC4R, 1470](#)
- [nppiSumGetBufferHostSize\\_16u\\_C1R, 1471](#)
- [nppiSumGetBufferHostSize\\_16u\\_C3R, 1471](#)
- [nppiSumGetBufferHostSize\\_16u\\_C4R, 1471](#)
- [nppiSumGetBufferHostSize\\_32f\\_AC4R, 1471](#)
- [nppiSumGetBufferHostSize\\_32f\\_C1R, 1472](#)
- [nppiSumGetBufferHostSize\\_32f\\_C3R, 1472](#)
- [nppiSumGetBufferHostSize\\_32f\\_C4R, 1472](#)
- [nppiSumGetBufferHostSize\\_8u64s\\_C1R, 1473](#)
- [nppiSumGetBufferHostSize\\_8u64s\\_C4R, 1473](#)
- [nppiSumGetBufferHostSize\\_8u\\_AC4R, 1473](#)
- [nppiSumGetBufferHostSize\\_8u\\_C1R, 1473](#)
- [nppiSumGetBufferHostSize\\_8u\\_C3R, 1474](#)
- [nppiSumGetBufferHostSize\\_8u\\_C4R, 1474](#)
- [image\\_swap\\_channels](#)
  - [nppiSwapChannels\\_16s\\_AC4R, 926](#)
  - [nppiSwapChannels\\_16s\\_C3C4R, 926](#)
  - [nppiSwapChannels\\_16s\\_C3IR, 926](#)
  - [nppiSwapChannels\\_16s\\_C3R, 927](#)
  - [nppiSwapChannels\\_16s\\_C4C3R, 927](#)
  - [nppiSwapChannels\\_16s\\_C4IR, 928](#)
  - [nppiSwapChannels\\_16s\\_C4R, 928](#)
  - [nppiSwapChannels\\_16u\\_AC4R, 928](#)
  - [nppiSwapChannels\\_16u\\_C3C4R, 929](#)
  - [nppiSwapChannels\\_16u\\_C3IR, 929](#)
  - [nppiSwapChannels\\_16u\\_C3R, 930](#)
  - [nppiSwapChannels\\_16u\\_C4C3R, 930](#)
  - [nppiSwapChannels\\_16u\\_C4IR, 931](#)
  - [nppiSwapChannels\\_16u\\_C4R, 931](#)
  - [nppiSwapChannels\\_32f\\_AC4R, 931](#)
  - [nppiSwapChannels\\_32f\\_C3C4R, 932](#)
  - [nppiSwapChannels\\_32f\\_C3IR, 932](#)
  - [nppiSwapChannels\\_32f\\_C3R, 933](#)
  - [nppiSwapChannels\\_32f\\_C4C3R, 933](#)
  - [nppiSwapChannels\\_32f\\_C4IR, 934](#)
  - [nppiSwapChannels\\_32f\\_C4R, 934](#)
  - [nppiSwapChannels\\_32s\\_AC4R, 934](#)
  - [nppiSwapChannels\\_32s\\_C3C4R, 935](#)
  - [nppiSwapChannels\\_32s\\_C3IR, 935](#)
  - [nppiSwapChannels\\_32s\\_C3R, 936](#)
  - [nppiSwapChannels\\_32s\\_C4C3R, 936](#)
  - [nppiSwapChannels\\_32s\\_C4IR, 937](#)
  - [nppiSwapChannels\\_32s\\_C4R, 937](#)
  - [nppiSwapChannels\\_8u\\_AC4R, 937](#)
  - [nppiSwapChannels\\_8u\\_C3C4R, 938](#)
  - [nppiSwapChannels\\_8u\\_C3IR, 938](#)
  - [nppiSwapChannels\\_8u\\_C3R, 939](#)
  - [nppiSwapChannels\\_8u\\_C4C3R, 939](#)
  - [nppiSwapChannels\\_8u\\_C4IR, 940](#)
  - [nppiSwapChannels\\_8u\\_C4R, 940](#)
- [image\\_threshold\\_operations](#)
  - [nppiThreshold\\_16s\\_AC4IR, 2145](#)
  - [nppiThreshold\\_16s\\_AC4R, 2145](#)
  - [nppiThreshold\\_16s\\_C1IR, 2146](#)
  - [nppiThreshold\\_16s\\_C1R, 2146](#)
  - [nppiThreshold\\_16s\\_C3IR, 2147](#)
  - [nppiThreshold\\_16s\\_C3R, 2147](#)
  - [nppiThreshold\\_16u\\_AC4IR, 2148](#)
  - [nppiThreshold\\_16u\\_AC4R, 2148](#)
  - [nppiThreshold\\_16u\\_C1IR, 2148](#)
  - [nppiThreshold\\_16u\\_C1R, 2149](#)
  - [nppiThreshold\\_16u\\_C3IR, 2149](#)
  - [nppiThreshold\\_16u\\_C3R, 2150](#)
  - [nppiThreshold\\_32f\\_AC4IR, 2150](#)
  - [nppiThreshold\\_32f\\_AC4R, 2151](#)
  - [nppiThreshold\\_32f\\_C1IR, 2151](#)
  - [nppiThreshold\\_32f\\_C1R, 2152](#)
  - [nppiThreshold\\_32f\\_C3IR, 2152](#)
  - [nppiThreshold\\_32f\\_C3R, 2152](#)
  - [nppiThreshold\\_8u\\_AC4IR, 2153](#)
  - [nppiThreshold\\_8u\\_AC4R, 2153](#)
  - [nppiThreshold\\_8u\\_C1IR, 2154](#)
  - [nppiThreshold\\_8u\\_C1R, 2154](#)
  - [nppiThreshold\\_8u\\_C3IR, 2155](#)
  - [nppiThreshold\\_8u\\_C3R, 2155](#)
  - [nppiThreshold\\_GT\\_16s\\_AC4IR, 2156](#)
  - [nppiThreshold\\_GT\\_16s\\_AC4R, 2156](#)
  - [nppiThreshold\\_GT\\_16s\\_C1IR, 2157](#)
  - [nppiThreshold\\_GT\\_16s\\_C1R, 2157](#)
  - [nppiThreshold\\_GT\\_16s\\_C3IR, 2157](#)
  - [nppiThreshold\\_GT\\_16s\\_C3R, 2158](#)
  - [nppiThreshold\\_GT\\_16u\\_AC4IR, 2158](#)
  - [nppiThreshold\\_GT\\_16u\\_AC4R, 2159](#)
  - [nppiThreshold\\_GT\\_16u\\_C1IR, 2159](#)
  - [nppiThreshold\\_GT\\_16u\\_C1R, 2159](#)
  - [nppiThreshold\\_GT\\_16u\\_C3IR, 2160](#)
  - [nppiThreshold\\_GT\\_16u\\_C3R, 2160](#)
  - [nppiThreshold\\_GT\\_32f\\_AC4IR, 2161](#)
  - [nppiThreshold\\_GT\\_32f\\_AC4R, 2161](#)
  - [nppiThreshold\\_GT\\_32f\\_C1IR, 2161](#)
  - [nppiThreshold\\_GT\\_32f\\_C1R, 2162](#)
  - [nppiThreshold\\_GT\\_32f\\_C3IR, 2162](#)



- [nppiThreshold\\_GT\\_32f\\_C3R, 2163](#)  
[nppiThreshold\\_GT\\_8u\\_AC4IR, 2163](#)  
[nppiThreshold\\_GT\\_8u\\_AC4R, 2163](#)  
[nppiThreshold\\_GT\\_8u\\_C1IR, 2164](#)  
[nppiThreshold\\_GT\\_8u\\_C1R, 2164](#)  
[nppiThreshold\\_GT\\_8u\\_C3IR, 2165](#)  
[nppiThreshold\\_GT\\_8u\\_C3R, 2165](#)  
[nppiThreshold\\_GTVVal\\_16s\\_AC4IR, 2165](#)  
[nppiThreshold\\_GTVVal\\_16s\\_AC4R, 2166](#)  
[nppiThreshold\\_GTVVal\\_16s\\_C1IR, 2166](#)  
[nppiThreshold\\_GTVVal\\_16s\\_C1R, 2167](#)  
[nppiThreshold\\_GTVVal\\_16s\\_C3IR, 2167](#)  
[nppiThreshold\\_GTVVal\\_16s\\_C3R, 2167](#)  
[nppiThreshold\\_GTVVal\\_16u\\_AC4IR, 2168](#)  
[nppiThreshold\\_GTVVal\\_16u\\_AC4R, 2168](#)  
[nppiThreshold\\_GTVVal\\_16u\\_C1IR, 2169](#)  
[nppiThreshold\\_GTVVal\\_16u\\_C1R, 2169](#)  
[nppiThreshold\\_GTVVal\\_16u\\_C3IR, 2170](#)  
[nppiThreshold\\_GTVVal\\_16u\\_C3R, 2170](#)  
[nppiThreshold\\_GTVVal\\_32f\\_AC4IR, 2170](#)  
[nppiThreshold\\_GTVVal\\_32f\\_AC4R, 2171](#)  
[nppiThreshold\\_GTVVal\\_32f\\_C1IR, 2171](#)  
[nppiThreshold\\_GTVVal\\_32f\\_C1R, 2172](#)  
[nppiThreshold\\_GTVVal\\_32f\\_C3IR, 2172](#)  
[nppiThreshold\\_GTVVal\\_32f\\_C3R, 2172](#)  
[nppiThreshold\\_GTVVal\\_8u\\_AC4IR, 2173](#)  
[nppiThreshold\\_GTVVal\\_8u\\_AC4R, 2173](#)  
[nppiThreshold\\_GTVVal\\_8u\\_C1IR, 2174](#)  
[nppiThreshold\\_GTVVal\\_8u\\_C1R, 2174](#)  
[nppiThreshold\\_GTVVal\\_8u\\_C3IR, 2175](#)  
[nppiThreshold\\_GTVVal\\_8u\\_C3R, 2175](#)  
[nppiThreshold\\_LT\\_16s\\_AC4IR, 2175](#)  
[nppiThreshold\\_LT\\_16s\\_AC4R, 2176](#)  
[nppiThreshold\\_LT\\_16s\\_C1IR, 2176](#)  
[nppiThreshold\\_LT\\_16s\\_C1R, 2177](#)  
[nppiThreshold\\_LT\\_16s\\_C3IR, 2177](#)  
[nppiThreshold\\_LT\\_16s\\_C3R, 2177](#)  
[nppiThreshold\\_LT\\_16u\\_AC4IR, 2178](#)  
[nppiThreshold\\_LT\\_16u\\_AC4R, 2178](#)  
[nppiThreshold\\_LT\\_16u\\_C1IR, 2179](#)  
[nppiThreshold\\_LT\\_16u\\_C1R, 2179](#)  
[nppiThreshold\\_LT\\_16u\\_C3IR, 2179](#)  
[nppiThreshold\\_LT\\_16u\\_C3R, 2180](#)  
[nppiThreshold\\_LT\\_32f\\_AC4IR, 2180](#)  
[nppiThreshold\\_LT\\_32f\\_AC4R, 2181](#)  
[nppiThreshold\\_LT\\_32f\\_C1IR, 2181](#)  
[nppiThreshold\\_LT\\_32f\\_C1R, 2181](#)  
[nppiThreshold\\_LT\\_32f\\_C3IR, 2182](#)  
[nppiThreshold\\_LT\\_32f\\_C3R, 2182](#)  
[nppiThreshold\\_LT\\_8u\\_AC4IR, 2183](#)  
[nppiThreshold\\_LT\\_8u\\_AC4R, 2183](#)  
[nppiThreshold\\_LT\\_8u\\_C1IR, 2183](#)  
[nppiThreshold\\_LT\\_8u\\_C1R, 2184](#)  
[nppiThreshold\\_LT\\_8u\\_C3IR, 2184](#)  
[nppiThreshold\\_LT\\_8u\\_C3R, 2185](#)  
[nppiThreshold\\_LTVVal\\_16s\\_AC4IR, 2185](#)  
[nppiThreshold\\_LTVVal\\_16s\\_AC4R, 2185](#)  
[nppiThreshold\\_LTVVal\\_16s\\_C1IR, 2186](#)  
[nppiThreshold\\_LTVVal\\_16s\\_C1R, 2186](#)  
[nppiThreshold\\_LTVVal\\_16s\\_C3IR, 2187](#)  
[nppiThreshold\\_LTVVal\\_16s\\_C3R, 2187](#)  
[nppiThreshold\\_LTVVal\\_16u\\_AC4IR, 2188](#)  
[nppiThreshold\\_LTVVal\\_16u\\_AC4R, 2188](#)  
[nppiThreshold\\_LTVVal\\_16u\\_C1IR, 2188](#)  
[nppiThreshold\\_LTVVal\\_16u\\_C1R, 2189](#)  
[nppiThreshold\\_LTVVal\\_16u\\_C3IR, 2189](#)  
[nppiThreshold\\_LTVVal\\_16u\\_C3R, 2190](#)  
[nppiThreshold\\_LTVVal\\_32f\\_AC4IR, 2190](#)  
[nppiThreshold\\_LTVVal\\_32f\\_AC4R, 2190](#)  
[nppiThreshold\\_LTVVal\\_32f\\_C1IR, 2191](#)  
[nppiThreshold\\_LTVVal\\_32f\\_C1R, 2191](#)  
[nppiThreshold\\_LTVVal\\_32f\\_C3IR, 2192](#)  
[nppiThreshold\\_LTVVal\\_32f\\_C3R, 2192](#)  
[nppiThreshold\\_LTVVal\\_8u\\_AC4IR, 2193](#)  
[nppiThreshold\\_LTVVal\\_8u\\_AC4R, 2193](#)  
[nppiThreshold\\_LTVVal\\_8u\\_C1IR, 2193](#)  
[nppiThreshold\\_LTVVal\\_8u\\_C1R, 2194](#)  
[nppiThreshold\\_LTVVal\\_8u\\_C3IR, 2194](#)  
[nppiThreshold\\_LTVVal\\_8u\\_C3R, 2195](#)  
[nppiThreshold\\_LTVValGTVVal\\_16s\\_AC4IR, 2195](#)  
[nppiThreshold\\_LTVValGTVVal\\_16s\\_AC4R, 2196](#)  
[nppiThreshold\\_LTVValGTVVal\\_16s\\_C1IR, 2196](#)  
[nppiThreshold\\_LTVValGTVVal\\_16s\\_C1R, 2197](#)  
[nppiThreshold\\_LTVValGTVVal\\_16s\\_C3IR, 2197](#)  
[nppiThreshold\\_LTVValGTVVal\\_16s\\_C3R, 2198](#)  
[nppiThreshold\\_LTVValGTVVal\\_16u\\_AC4IR, 2198](#)  
[nppiThreshold\\_LTVValGTVVal\\_16u\\_AC4R, 2199](#)  
[nppiThreshold\\_LTVValGTVVal\\_16u\\_C1IR, 2199](#)  
[nppiThreshold\\_LTVValGTVVal\\_16u\\_C1R, 2200](#)  
[nppiThreshold\\_LTVValGTVVal\\_16u\\_C3IR, 2200](#)  
[nppiThreshold\\_LTVValGTVVal\\_16u\\_C3R, 2201](#)  
[nppiThreshold\\_LTVValGTVVal\\_32f\\_AC4IR, 2201](#)  
[nppiThreshold\\_LTVValGTVVal\\_32f\\_AC4R, 2202](#)  
[nppiThreshold\\_LTVValGTVVal\\_32f\\_C1IR, 2202](#)  
[nppiThreshold\\_LTVValGTVVal\\_32f\\_C1R, 2203](#)  
[nppiThreshold\\_LTVValGTVVal\\_32f\\_C3IR, 2203](#)  
[nppiThreshold\\_LTVValGTVVal\\_32f\\_C3R, 2204](#)  
[nppiThreshold\\_LTVValGTVVal\\_8u\\_AC4IR, 2204](#)  
[nppiThreshold\\_LTVValGTVVal\\_8u\\_AC4R, 2205](#)  
[nppiThreshold\\_LTVValGTVVal\\_8u\\_C1IR, 2205](#)  
[nppiThreshold\\_LTVValGTVVal\\_8u\\_C1R, 2206](#)

- [nppiThreshold\\_LTValGTVal\\_8u\\_C3IR](#), 2206
- [nppiThreshold\\_LTValGTVal\\_8u\\_C3R](#), 2207
- [nppiThreshold\\_Val\\_16s\\_AC4IR](#), 2207
- [nppiThreshold\\_Val\\_16s\\_AC4R](#), 2208
- [nppiThreshold\\_Val\\_16s\\_C1IR](#), 2208
- [nppiThreshold\\_Val\\_16s\\_C1R](#), 2209
- [nppiThreshold\\_Val\\_16s\\_C3IR](#), 2209
- [nppiThreshold\\_Val\\_16s\\_C3R](#), 2210
- [nppiThreshold\\_Val\\_16u\\_AC4IR](#), 2210
- [nppiThreshold\\_Val\\_16u\\_AC4R](#), 2211
- [nppiThreshold\\_Val\\_16u\\_C1IR](#), 2211
- [nppiThreshold\\_Val\\_16u\\_C1R](#), 2212
- [nppiThreshold\\_Val\\_16u\\_C3IR](#), 2212
- [nppiThreshold\\_Val\\_16u\\_C3R](#), 2213
- [nppiThreshold\\_Val\\_32f\\_AC4IR](#), 2213
- [nppiThreshold\\_Val\\_32f\\_AC4R](#), 2214
- [nppiThreshold\\_Val\\_32f\\_C1IR](#), 2214
- [nppiThreshold\\_Val\\_32f\\_C1R](#), 2215
- [nppiThreshold\\_Val\\_32f\\_C3IR](#), 2215
- [nppiThreshold\\_Val\\_32f\\_C3R](#), 2216
- [nppiThreshold\\_Val\\_8u\\_AC4IR](#), 2216
- [nppiThreshold\\_Val\\_8u\\_AC4R](#), 2217
- [nppiThreshold\\_Val\\_8u\\_C1IR](#), 2217
- [nppiThreshold\\_Val\\_8u\\_C1R](#), 2218
- [nppiThreshold\\_Val\\_8u\\_C3IR](#), 2218
- [nppiThreshold\\_Val\\_8u\\_C3R](#), 2219
- [image\\_transpose](#)
  - [nppiTranspose\\_16s\\_C1R](#), 917
  - [nppiTranspose\\_16s\\_C3R](#), 917
  - [nppiTranspose\\_16s\\_C4R](#), 918
  - [nppiTranspose\\_16u\\_C1R](#), 918
  - [nppiTranspose\\_16u\\_C3R](#), 918
  - [nppiTranspose\\_16u\\_C4R](#), 919
  - [nppiTranspose\\_32f\\_C1R](#), 919
  - [nppiTranspose\\_32f\\_C3R](#), 919
  - [nppiTranspose\\_32f\\_C4R](#), 920
  - [nppiTranspose\\_32s\\_C1R](#), 920
  - [nppiTranspose\\_32s\\_C3R](#), 920
  - [nppiTranspose\\_32s\\_C4R](#), 921
  - [nppiTranspose\\_8u\\_C1R](#), 921
  - [nppiTranspose\\_8u\\_C3R](#), 921
  - [nppiTranspose\\_8u\\_C4R](#), 922
- [image\\_xor](#)
  - [nppiXor\\_16u\\_AC4IR](#), 458
  - [nppiXor\\_16u\\_AC4R](#), 458
  - [nppiXor\\_16u\\_C1IR](#), 458
  - [nppiXor\\_16u\\_C1R](#), 459
  - [nppiXor\\_16u\\_C3IR](#), 459
  - [nppiXor\\_16u\\_C3R](#), 459
  - [nppiXor\\_16u\\_C4IR](#), 460
  - [nppiXor\\_16u\\_C4R](#), 460
  - [nppiXor\\_32s\\_AC4IR](#), 461
  - [nppiXor\\_32s\\_AC4R](#), 461
  - [nppiXor\\_32s\\_C1IR](#), 461
  - [nppiXor\\_32s\\_C1R](#), 462
  - [nppiXor\\_32s\\_C3IR](#), 462
  - [nppiXor\\_32s\\_C3R](#), 462
  - [nppiXor\\_32s\\_C4IR](#), 463
  - [nppiXor\\_32s\\_C4R](#), 463
  - [nppiXor\\_8u\\_AC4IR](#), 464
  - [nppiXor\\_8u\\_AC4R](#), 464
  - [nppiXor\\_8u\\_C1IR](#), 464
  - [nppiXor\\_8u\\_C1R](#), 465
  - [nppiXor\\_8u\\_C3IR](#), 465
  - [nppiXor\\_8u\\_C3R](#), 465
  - [nppiXor\\_8u\\_C4IR](#), 466
  - [nppiXor\\_8u\\_C4R](#), 466
- [image\\_xorc](#)
  - [nppiXorC\\_16u\\_AC4IR](#), 395
  - [nppiXorC\\_16u\\_AC4R](#), 395
  - [nppiXorC\\_16u\\_C1IR](#), 395
  - [nppiXorC\\_16u\\_C1R](#), 396
  - [nppiXorC\\_16u\\_C3IR](#), 396
  - [nppiXorC\\_16u\\_C3R](#), 396
  - [nppiXorC\\_16u\\_C4IR](#), 397
  - [nppiXorC\\_16u\\_C4R](#), 397
  - [nppiXorC\\_32s\\_AC4IR](#), 397
  - [nppiXorC\\_32s\\_AC4R](#), 398
  - [nppiXorC\\_32s\\_C1IR](#), 398
  - [nppiXorC\\_32s\\_C1R](#), 398
  - [nppiXorC\\_32s\\_C3IR](#), 399
  - [nppiXorC\\_32s\\_C3R](#), 399
  - [nppiXorC\\_32s\\_C4IR](#), 399
  - [nppiXorC\\_32s\\_C4R](#), 400
  - [nppiXorC\\_8u\\_AC4IR](#), 400
  - [nppiXorC\\_8u\\_AC4R](#), 400
  - [nppiXorC\\_8u\\_C1IR](#), 401
  - [nppiXorC\\_8u\\_C1R](#), 401
  - [nppiXorC\\_8u\\_C3IR](#), 401
  - [nppiXorC\\_8u\\_C3R](#), 402
  - [nppiXorC\\_8u\\_C4IR](#), 402
  - [nppiXorC\\_8u\\_C4R](#), 402
- [Infinity Norm](#), 2514
- [Infinity Norm Diff](#), 2531
- [Initialization](#), 2443
- [Integral](#), 1846, 2442
- [L1 Norm](#), 2519
- [L1 Norm Diff](#), 2536
- [L2 Norm](#), 2525
- [L2 Norm Diff](#), 2542
- [Labeling and Segmentation](#), 711
- [Linear Transforms](#), 1363
- [Ln](#), 356, 2366
- [Logical And Shift Operations](#), 2382
- [Logical Operations](#), 370
- [LShiftC](#), 421, 2404

- major
  - NppLibraryVersion, 2634
- Malloc, 2618
- Max, 1502
- MaxEvery, 1832
- Maximum, 2469
- MaximumError, 2024, 2571
- MaximumRelativeError, 2070, 2593
- MaxIndx, 1515
- Mean, 1560, 2489
- Mean And Standard Deviation, 2498
- Mean\_StdDev, 1581
- Memory Management, 2118, 2617
- Min, 1475
- MinEvery, 1839
- MinEvery And MaxEvery Functions, 2458
- Minimum, 2479
- Minimum\_Maximum, 2502
- MinIndx, 1488
- MinMax, 1529
- MinMaxIndx, 1543
- minor
  - NppLibraryVersion, 2634
- Mirror, 1250
- Morphological Operations, 1366
- Mul, 208, 2310
- MulC, 81, 2257
- MulCScale, 107
- MulScale, 237
  
- Norm\_Inf, 1599
- Norm\_L1, 1621
- Norm\_L2, 1642
- Normalize, 2377
- NormDiff\_Inf, 1663
- NormDiff\_L1, 1686
- NormDiff\_L2, 1709
- NormRel\_Inf, 1732
- NormRel\_L1, 1755
- NormRel\_L2, 1778
- Not, 468, 2401
- NPP Core, 31
- NPP Image Processing, 51
- NPP Signal Processing, 2243
- NPP Type Definitions and Constants, 34
- Npp16s
  - npp\_basic\_types, 48
- Npp16sc
  - npp\_basic\_types, 50
- Npp16u
  - npp\_basic\_types, 48
- Npp16uc
  - npp\_basic\_types, 50
- Npp32f
  - npp\_basic\_types, 48
- Npp32fc
  - npp\_basic\_types, 48
- Npp32s
  - npp\_basic\_types, 48
- Npp32sc
  - npp\_basic\_types, 48
- Npp32u
  - npp\_basic\_types, 49
- Npp32uc
  - npp\_basic\_types, 49
- Npp64f
  - npp\_basic\_types, 49
- Npp64fc
  - npp\_basic\_types, 49
- Npp64s
  - npp\_basic\_types, 49
- Npp64sc
  - npp\_basic\_types, 49
- Npp64u
  - npp\_basic\_types, 49
- Npp8s
  - npp\_basic\_types, 49
- Npp8u
  - npp\_basic\_types, 49
- Npp8uc
  - npp\_basic\_types, 50
- NPP\_AFFINE\_QUAD\_INCORRECT\_WARNING
  - typedefs\_npp, 46
- NPP\_ALG\_HINT\_ACCURATE
  - typedefs\_npp, 41
- NPP\_ALG\_HINT\_FAST
  - typedefs\_npp, 41
- NPP\_ALG\_HINT\_NONE
  - typedefs\_npp, 41
- NPP\_ALIGNMENT\_ERROR
  - typedefs\_npp, 44
- NPP\_ANCHOR\_ERROR
  - typedefs\_npp, 45
- NPP\_BAD\_ARGUMENT\_ERROR
  - typedefs\_npp, 45
- NPP\_BORDER\_CONSTANT
  - typedefs\_npp, 42
- NPP\_BORDER\_NONE
  - typedefs\_npp, 42
- NPP\_BORDER\_REPLICATE
  - typedefs\_npp, 42
- NPP\_BORDER\_UNDEFINED
  - typedefs\_npp, 42
- NPP\_BORDER\_WRAP
  - typedefs\_npp, 42
- NPP\_BOTH\_AXIS
  - typedefs\_npp, 42
- NPP\_CHANNEL\_ERROR



- typedefs\_npp, 45
- NPP\_CHANNEL\_ORDER\_ERROR
  - typedefs\_npp, 45
- NPP\_CMP\_EQ
  - typedefs\_npp, 41
- NPP\_CMP\_GREATER
  - typedefs\_npp, 41
- NPP\_CMP\_GREATER\_EQ
  - typedefs\_npp, 41
- NPP\_CMP\_LESS
  - typedefs\_npp, 41
- NPP\_CMP\_LESS\_EQ
  - typedefs\_npp, 41
- NPP\_COEFFICIENT\_ERROR
  - typedefs\_npp, 45
- NPP\_COI\_ERROR
  - typedefs\_npp, 45
- NPP\_CONTEXT\_MATCH\_ERROR
  - typedefs\_npp, 45
- NPP\_CUDA\_1\_0
  - typedefs\_npp, 41
- NPP\_CUDA\_1\_1
  - typedefs\_npp, 41
- NPP\_CUDA\_1\_2
  - typedefs\_npp, 41
- NPP\_CUDA\_1\_3
  - typedefs\_npp, 41
- NPP\_CUDA\_2\_0
  - typedefs\_npp, 41
- NPP\_CUDA\_2\_1
  - typedefs\_npp, 41
- NPP\_CUDA\_3\_0
  - typedefs\_npp, 41
- NPP\_CUDA\_3\_2
  - typedefs\_npp, 41
- NPP\_CUDA\_3\_5
  - typedefs\_npp, 41
- NPP\_CUDA\_5\_0
  - typedefs\_npp, 41
- NPP\_CUDA\_KERNEL\_EXECUTION\_ERROR
  - typedefs\_npp, 44
- NPP\_CUDA\_NOT\_CAPABLE
  - typedefs\_npp, 41
- NPP\_CUDA\_UNKNOWN\_VERSION
  - typedefs\_npp, 41
- NPP\_DATA\_TYPE\_ERROR
  - typedefs\_npp, 45
- NPP\_DIVIDE\_BY\_ZERO\_ERROR
  - typedefs\_npp, 45
- NPP\_DIVIDE\_BY\_ZERO\_WARNING
  - typedefs\_npp, 46
- NPP\_DIVISOR\_ERROR
  - typedefs\_npp, 45
- NPP\_DOUBLE\_SIZE\_WARNING
  - typedefs\_npp, 45
- typedefs\_npp, 46
- NPP\_ERROR
  - typedefs\_npp, 45
- NPP\_ERROR\_RESERVED
  - typedefs\_npp, 45
- NPP\_FFT\_FLAG\_ERROR
  - typedefs\_npp, 45
- NPP\_FFT\_ORDER\_ERROR
  - typedefs\_npp, 45
- NPP\_HAAR\_CLASSIFIER\_PIXEL\_MATCH\_ERROR
  - typedefs\_npp, 44
- NPP\_HISTOGRAM\_NUMBER\_OF\_LEVELS\_ERROR
  - typedefs\_npp, 45
- NPP\_HORIZONTAL\_AXIS
  - typedefs\_npp, 42
- NPP\_INTERPOLATION\_ERROR
  - typedefs\_npp, 45
- NPP\_INVALID\_DEVICE\_POINTER\_ERROR
  - typedefs\_npp, 44
- NPP\_INVALID\_HOST\_POINTER\_ERROR
  - typedefs\_npp, 44
- NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR
  - typedefs\_npp, 45
- NPP\_LUT\_PALETTE\_BITSIZE\_ERROR
  - typedefs\_npp, 44
- NPP\_MASK\_SIZE\_1\_X\_3
  - typedefs\_npp, 43
- NPP\_MASK\_SIZE\_1\_X\_5
  - typedefs\_npp, 43
- NPP\_MASK\_SIZE\_3\_X\_1
  - typedefs\_npp, 43
- NPP\_MASK\_SIZE\_3\_X\_3
  - typedefs\_npp, 43
- NPP\_MASK\_SIZE\_5\_X\_1
  - typedefs\_npp, 43
- NPP\_MASK\_SIZE\_5\_X\_5
  - typedefs\_npp, 43
- NPP\_MASK\_SIZE\_ERROR
  - typedefs\_npp, 45
- NPP\_MEMCPY\_ERROR
  - typedefs\_npp, 44
- NPP\_MEMFREE\_ERROR
  - typedefs\_npp, 44
- NPP\_MEMORY\_ALLOCATION\_ERR
  - typedefs\_npp, 45
- NPP\_MEMSET\_ERROR
  - typedefs\_npp, 44
- NPP\_MIRROR\_FLIP\_ERROR
  - typedefs\_npp, 45
- NPP\_MISALIGNED\_DST\_ROI\_WARNING
  - typedefs\_npp, 46
- NPP\_MOMENT\_00\_ZERO\_ERROR

- typedefs\_npp, 45
- NPP\_NO\_ERROR
  - typedefs\_npp, 45
- NPP\_NO\_MEMORY\_ERROR
  - typedefs\_npp, 45
- NPP\_NO\_OPERATION\_WARNING
  - typedefs\_npp, 46
- NPP\_NOT\_EVEN\_STEP\_ERROR
  - typedefs\_npp, 45
- NPP\_NOT\_IMPLEMENTED\_ERROR
  - typedefs\_npp, 45
- NPP\_NOT\_SUFFICIENT\_COMPUTE\_ -  
CAPABILITY
  - typedefs\_npp, 44
- NPP\_NOT\_SUPPORTED\_MODE\_ERROR
  - typedefs\_npp, 44
- NPP\_NULL\_POINTER\_ERROR
  - typedefs\_npp, 45
- NPP\_NUMBER\_OF\_CHANNELS\_ERROR
  - typedefs\_npp, 45
- NPP\_OUT\_OFF\_RANGE\_ERROR
  - typedefs\_npp, 45
- NPP\_QUADRANGLE\_ERROR
  - typedefs\_npp, 45
- NPP\_QUALITY\_INDEX\_ERROR
  - typedefs\_npp, 44
- NPP\_RANGE\_ERROR
  - typedefs\_npp, 45
- NPP\_RECTANGLE\_ERROR
  - typedefs\_npp, 45
- NPP\_RESIZE\_FACTOR\_ERROR
  - typedefs\_npp, 45
- NPP\_RESIZE\_NO\_OPERATION\_ERROR
  - typedefs\_npp, 44
- NPP\_RND\_FINANCIAL
  - typedefs\_npp, 44
- NPP\_RND\_NEAR
  - typedefs\_npp, 43
- NPP\_RND\_ZERO
  - typedefs\_npp, 44
- NPP\_ROUND\_MODE\_NOT\_SUPPORTED\_ -  
ERROR
  - typedefs\_npp, 44
- NPP\_ROUND\_NEAREST\_TIES\_AWAY\_ -  
FROM\_ZERO
  - typedefs\_npp, 44
- NPP\_ROUND\_NEAREST\_TIES\_TO\_EVEN
  - typedefs\_npp, 44
- NPP\_ROUND\_TOWARD\_ZERO
  - typedefs\_npp, 44
- NPP\_SCALE\_RANGE\_ERROR
  - typedefs\_npp, 45
- NPP\_SIZE\_ERROR
  - typedefs\_npp, 45
- NPP\_STEP\_ERROR
  - typedefs\_npp, 45
- NPP\_STRIDE\_ERROR
  - typedefs\_npp, 45
- NPP\_SUCCESS
  - typedefs\_npp, 46
- NPP\_TEXTURE\_BIND\_ERROR
  - typedefs\_npp, 44
- NPP\_THRESHOLD\_ERROR
  - typedefs\_npp, 45
- NPP\_THRESHOLD\_NEGATIVE\_LEVEL\_ -  
ERROR
  - typedefs\_npp, 45
- NPP\_VERTICAL\_AXIS
  - typedefs\_npp, 42
- NPP\_WRONG\_INTERSECTION\_QUAD\_ -  
WARNING
  - typedefs\_npp, 46
- NPP\_WRONG\_INTERSECTION\_ROI\_ERROR
  - typedefs\_npp, 44
- NPP\_WRONG\_INTERSECTION\_ROI\_ -  
WARNING
  - typedefs\_npp, 46
- NPP\_ZC\_MODE\_NOT\_SUPPORTED\_ERROR
  - typedefs\_npp, 44
- NPP\_ZERO\_MASK\_VALUE\_ERROR
  - typedefs\_npp, 45
- NPP\_ALIGN\_16, 2625
  - im, 2625
  - re, 2626
- NPP\_ALIGN\_8, 2627
  - im, 2627
  - re, 2627, 2628
- npp\_basic\_types
  - \_\_align\_\_, 49, 50
  - Npp16s, 48
  - Npp16sc, 50
  - Npp16u, 48
  - Npp16uc, 50
  - Npp32f, 48
  - Npp32fc, 48
  - Npp32s, 48
  - Npp32sc, 48
  - Npp32u, 49
  - Npp32uc, 49
  - Npp64f, 49
  - Npp64fc, 49
  - Npp64s, 49
  - Npp64sc, 49
  - Npp64u, 49
  - Npp8s, 49
  - Npp8u, 49
  - Npp8uc, 50
- NPP\_MAX\_16S

- typedefs\_npp, 39
- NPP\_MAX\_16U
  - typedefs\_npp, 39
- NPP\_MAX\_32S
  - typedefs\_npp, 39
- NPP\_MAX\_32U
  - typedefs\_npp, 39
- NPP\_MAX\_64S
  - typedefs\_npp, 39
- NPP\_MAX\_64U
  - typedefs\_npp, 39
- NPP\_MAX\_8S
  - typedefs\_npp, 39
- NPP\_MAX\_8U
  - typedefs\_npp, 40
- NPP\_MAXABS\_32F
  - typedefs\_npp, 40
- NPP\_MAXABS\_64F
  - typedefs\_npp, 40
- NPP\_MIN\_16S
  - typedefs\_npp, 40
- NPP\_MIN\_16U
  - typedefs\_npp, 40
- NPP\_MIN\_32S
  - typedefs\_npp, 40
- NPP\_MIN\_32U
  - typedefs\_npp, 40
- NPP\_MIN\_64S
  - typedefs\_npp, 40
- NPP\_MIN\_64U
  - typedefs\_npp, 40
- NPP\_MIN\_8S
  - typedefs\_npp, 40
- NPP\_MIN\_8U
  - typedefs\_npp, 40
- NPP\_MINABS\_32F
  - typedefs\_npp, 41
- NPP\_MINABS\_64F
  - typedefs\_npp, 41
- NppCmpOp
  - typedefs\_npp, 41
- nppGetGpuComputeCapability
  - core\_npp, 32
- nppGetGpuName
  - core\_npp, 32
- nppGetGpuNumSMs
  - core\_npp, 32
- nppGetLibVersion
  - core\_npp, 32
- nppGetMaxThreadsPerBlock
  - core\_npp, 32
- nppGetMaxThreadsPerSM
  - core\_npp, 32
- nppGetStream
  - core\_npp, 33
- NppGpuComputeCapability
  - typedefs\_npp, 41
- NppHintAlgorithm
  - typedefs\_npp, 41
- NPPI\_INTER\_CUBIC
  - typedefs\_npp, 43
- NPPI\_INTER\_CUBIC2P\_B05C03
  - typedefs\_npp, 43
- NPPI\_INTER\_CUBIC2P\_BSPLINE
  - typedefs\_npp, 43
- NPPI\_INTER\_CUBIC2P\_CATMULLROM
  - typedefs\_npp, 43
- NPPI\_INTER\_LANCZOS
  - typedefs\_npp, 43
- NPPI\_INTER\_LINEAR
  - typedefs\_npp, 43
- NPPI\_INTER\_NN
  - typedefs\_npp, 43
- NPPI\_INTER\_SUPER
  - typedefs\_npp, 43
- NPPI\_INTER\_UNDEFINED
  - typedefs\_npp, 43
- NPPI\_OP\_ALPHA\_ATOP
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_ATOP\_PREMUL
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_IN
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_IN\_PREMUL
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_OUT
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_OUT\_PREMUL
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_OVER
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_OVER\_PREMUL
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_PLUS
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_PLUS\_PREMUL
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_PREMUL
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_XOR
  - typedefs\_npp, 42
- NPPI\_OP\_ALPHA\_XOR\_PREMUL
  - typedefs\_npp, 42
- NPPI\_SMOOTH\_EDGE
  - typedefs\_npp, 43
- nppiAbs\_16s\_AC4IR
  - image\_abs, 321
- nppiAbs\_16s\_AC4R

image\_abs, [321](#)  
 nppiAbs\_16s\_C1IR  
   image\_abs, [321](#)  
 nppiAbs\_16s\_C1R  
   image\_abs, [322](#)  
 nppiAbs\_16s\_C3IR  
   image\_abs, [322](#)  
 nppiAbs\_16s\_C3R  
   image\_abs, [322](#)  
 nppiAbs\_16s\_C4IR  
   image\_abs, [323](#)  
 nppiAbs\_16s\_C4R  
   image\_abs, [323](#)  
 nppiAbs\_32f\_AC4IR  
   image\_abs, [323](#)  
 nppiAbs\_32f\_AC4R  
   image\_abs, [324](#)  
 nppiAbs\_32f\_C1IR  
   image\_abs, [324](#)  
 nppiAbs\_32f\_C1R  
   image\_abs, [324](#)  
 nppiAbs\_32f\_C3IR  
   image\_abs, [325](#)  
 nppiAbs\_32f\_C3R  
   image\_abs, [325](#)  
 nppiAbs\_32f\_C4IR  
   image\_abs, [325](#)  
 nppiAbs\_32f\_C4R  
   image\_abs, [326](#)  
 nppiAbsDiff\_16u\_C1R  
   image\_absdiff, [327](#)  
 nppiAbsDiff\_32f\_C1R  
   image\_absdiff, [328](#)  
 nppiAbsDiff\_8u\_C1R  
   image\_absdiff, [328](#)  
 nppiAbsDiff\_8u\_C3R  
   image\_absdiff, [328](#)  
 nppiAbsDiff\_8u\_C4R  
   image\_absdiff, [329](#)  
 nppiAbsDiffC\_16u\_C1R  
   image\_absdiffc, [166](#)  
 nppiAbsDiffC\_32f\_C1R  
   image\_absdiffc, [166](#)  
 nppiAbsDiffC\_8u\_C1R  
   image\_absdiffc, [167](#)  
 nppiACTable  
   typedefs\_npp, [42](#)  
 nppiAdd\_16s\_AC4IRSfs  
   image\_add, [173](#)  
 nppiAdd\_16s\_AC4RSfs  
   image\_add, [173](#)  
 nppiAdd\_16s\_C1IRSfs  
   image\_add, [174](#)  
 nppiAdd\_16s\_C1RSfs  
   image\_add, [174](#)  
 nppiAdd\_16s\_C3IRSfs  
   image\_add, [175](#)  
 nppiAdd\_16s\_C3RSfs  
   image\_add, [175](#)  
 nppiAdd\_16s\_C4IRSfs  
   image\_add, [175](#)  
 nppiAdd\_16s\_C4RSfs  
   image\_add, [176](#)  
 nppiAdd\_16sc\_AC4IRSfs  
   image\_add, [176](#)  
 nppiAdd\_16sc\_AC4RSfs  
   image\_add, [177](#)  
 nppiAdd\_16sc\_C1IRSfs  
   image\_add, [177](#)  
 nppiAdd\_16sc\_C1RSfs  
   image\_add, [177](#)  
 nppiAdd\_16sc\_C3IRSfs  
   image\_add, [178](#)  
 nppiAdd\_16sc\_C3RSfs  
   image\_add, [178](#)  
 nppiAdd\_16u\_AC4IRSfs  
   image\_add, [179](#)  
 nppiAdd\_16u\_AC4RSfs  
   image\_add, [179](#)  
 nppiAdd\_16u\_C1IRSfs  
   image\_add, [180](#)  
 nppiAdd\_16u\_C1RSfs  
   image\_add, [180](#)  
 nppiAdd\_16u\_C3IRSfs  
   image\_add, [180](#)  
 nppiAdd\_16u\_C3RSfs  
   image\_add, [181](#)  
 nppiAdd\_16u\_C4IRSfs  
   image\_add, [181](#)  
 nppiAdd\_16u\_C4RSfs  
   image\_add, [182](#)  
 nppiAdd\_32f\_AC4IR  
   image\_add, [182](#)  
 nppiAdd\_32f\_AC4R  
   image\_add, [182](#)  
 nppiAdd\_32f\_C1IR  
   image\_add, [183](#)  
 nppiAdd\_32f\_C1R  
   image\_add, [183](#)  
 nppiAdd\_32f\_C3IR  
   image\_add, [184](#)  
 nppiAdd\_32f\_C3R  
   image\_add, [184](#)  
 nppiAdd\_32f\_C4IR  
   image\_add, [184](#)  
 nppiAdd\_32f\_C4R  
   image\_add, [185](#)  
 nppiAdd\_32fc\_AC4IR

- image\_add, [185](#)
- nppiAdd\_32fc\_AC4R
  - image\_add, [185](#)
- nppiAdd\_32fc\_C1IR
  - image\_add, [186](#)
- nppiAdd\_32fc\_C1R
  - image\_add, [186](#)
- nppiAdd\_32fc\_C3IR
  - image\_add, [187](#)
- nppiAdd\_32fc\_C3R
  - image\_add, [187](#)
- nppiAdd\_32fc\_C4IR
  - image\_add, [187](#)
- nppiAdd\_32fc\_C4R
  - image\_add, [188](#)
- nppiAdd\_32s\_C1IRSfs
  - image\_add, [188](#)
- nppiAdd\_32s\_C1R
  - image\_add, [189](#)
- nppiAdd\_32s\_C1RSfs
  - image\_add, [189](#)
- nppiAdd\_32s\_C3IRSfs
  - image\_add, [189](#)
- nppiAdd\_32s\_C3RSfs
  - image\_add, [190](#)
- nppiAdd\_32sc\_AC4IRSfs
  - image\_add, [190](#)
- nppiAdd\_32sc\_AC4RSfs
  - image\_add, [191](#)
- nppiAdd\_32sc\_C1IRSfs
  - image\_add, [191](#)
- nppiAdd\_32sc\_C1RSfs
  - image\_add, [191](#)
- nppiAdd\_32sc\_C3IRSfs
  - image\_add, [192](#)
- nppiAdd\_32sc\_C3RSfs
  - image\_add, [192](#)
- nppiAdd\_8u\_AC4IRSfs
  - image\_add, [193](#)
- nppiAdd\_8u\_AC4RSfs
  - image\_add, [193](#)
- nppiAdd\_8u\_C1IRSfs
  - image\_add, [194](#)
- nppiAdd\_8u\_C1RSfs
  - image\_add, [194](#)
- nppiAdd\_8u\_C3IRSfs
  - image\_add, [194](#)
- nppiAdd\_8u\_C3RSfs
  - image\_add, [195](#)
- nppiAdd\_8u\_C4IRSfs
  - image\_add, [195](#)
- nppiAdd\_8u\_C4RSfs
  - image\_add, [196](#)
- nppiAddC\_16s\_AC4IRSfs
  - image\_addc, [60](#)
- nppiAddC\_16s\_AC4RSfs
  - image\_addc, [60](#)
- nppiAddC\_16s\_C1IRSfs
  - image\_addc, [60](#)
- nppiAddC\_16s\_C1RSfs
  - image\_addc, [61](#)
- nppiAddC\_16s\_C3IRSfs
  - image\_addc, [61](#)
- nppiAddC\_16s\_C3RSfs
  - image\_addc, [61](#)
- nppiAddC\_16s\_C4IRSfs
  - image\_addc, [62](#)
- nppiAddC\_16s\_C4RSfs
  - image\_addc, [62](#)
- nppiAddC\_16sc\_AC4IRSfs
  - image\_addc, [63](#)
- nppiAddC\_16sc\_AC4RSfs
  - image\_addc, [63](#)
- nppiAddC\_16sc\_C1IRSfs
  - image\_addc, [63](#)
- nppiAddC\_16sc\_C1RSfs
  - image\_addc, [64](#)
- nppiAddC\_16sc\_C3IRSfs
  - image\_addc, [64](#)
- nppiAddC\_16sc\_C3RSfs
  - image\_addc, [65](#)
- nppiAddC\_16u\_AC4IRSfs
  - image\_addc, [65](#)
- nppiAddC\_16u\_AC4RSfs
  - image\_addc, [65](#)
- nppiAddC\_16u\_C1IRSfs
  - image\_addc, [66](#)
- nppiAddC\_16u\_C1RSfs
  - image\_addc, [66](#)
- nppiAddC\_16u\_C3IRSfs
  - image\_addc, [67](#)
- nppiAddC\_16u\_C3RSfs
  - image\_addc, [67](#)
- nppiAddC\_16u\_C4IRSfs
  - image\_addc, [67](#)
- nppiAddC\_16u\_C4RSfs
  - image\_addc, [68](#)
- nppiAddC\_32f\_AC4IR
  - image\_addc, [68](#)
- nppiAddC\_32f\_AC4R
  - image\_addc, [68](#)
- nppiAddC\_32f\_C1IR
  - image\_addc, [69](#)
- nppiAddC\_32f\_C1R
  - image\_addc, [69](#)
- nppiAddC\_32f\_C3IR
  - image\_addc, [69](#)
- nppiAddC\_32f\_C3R

- image\_addc, [70](#)
- npAddC\_32f\_C4IR
  - image\_addc, [70](#)
- npAddC\_32f\_C4R
  - image\_addc, [70](#)
- npAddC\_32fc\_AC4IR
  - image\_addc, [71](#)
- npAddC\_32fc\_AC4R
  - image\_addc, [71](#)
- npAddC\_32fc\_C1IR
  - image\_addc, [71](#)
- npAddC\_32fc\_C1R
  - image\_addc, [72](#)
- npAddC\_32fc\_C3IR
  - image\_addc, [72](#)
- npAddC\_32fc\_C3R
  - image\_addc, [72](#)
- npAddC\_32fc\_C4IR
  - image\_addc, [73](#)
- npAddC\_32fc\_C4R
  - image\_addc, [73](#)
- npAddC\_32s\_C1IRSfs
  - image\_addc, [74](#)
- npAddC\_32s\_C1RSfs
  - image\_addc, [74](#)
- npAddC\_32s\_C3IRSfs
  - image\_addc, [74](#)
- npAddC\_32s\_C3RSfs
  - image\_addc, [75](#)
- npAddC\_32sc\_AC4IRSfs
  - image\_addc, [75](#)
- npAddC\_32sc\_AC4RSfs
  - image\_addc, [75](#)
- npAddC\_32sc\_C1IRSfs
  - image\_addc, [76](#)
- npAddC\_32sc\_C1RSfs
  - image\_addc, [76](#)
- npAddC\_32sc\_C3IRSfs
  - image\_addc, [77](#)
- npAddC\_32sc\_C3RSfs
  - image\_addc, [77](#)
- npAddC\_8u\_AC4IRSfs
  - image\_addc, [77](#)
- npAddC\_8u\_AC4RSfs
  - image\_addc, [78](#)
- npAddC\_8u\_C1IRSfs
  - image\_addc, [78](#)
- npAddC\_8u\_C1RSfs
  - image\_addc, [79](#)
- npAddC\_8u\_C3IRSfs
  - image\_addc, [79](#)
- npAddC\_8u\_C3RSfs
  - image\_addc, [79](#)
- npAddC\_8u\_C4IRSfs
  - image\_addc, [80](#)
- npAddC\_8u\_C4RSfs
  - image\_addc, [80](#)
- npAddProduct\_16u32f\_C1IMR
  - image\_addproduct, [200](#)
- npAddProduct\_16u32f\_C1IR
  - image\_addproduct, [201](#)
- npAddProduct\_32f\_C1IMR
  - image\_addproduct, [201](#)
- npAddProduct\_32f\_C1IR
  - image\_addproduct, [202](#)
- npAddProduct\_8u32f\_C1IMR
  - image\_addproduct, [202](#)
- npAddProduct\_8u32f\_C1IR
  - image\_addproduct, [202](#)
- npAddSquare\_16u32f\_C1IMR
  - image\_addsquare, [197](#)
- npAddSquare\_16u32f\_C1IR
  - image\_addsquare, [198](#)
- npAddSquare\_32f\_C1IMR
  - image\_addsquare, [198](#)
- npAddSquare\_32f\_C1IR
  - image\_addsquare, [198](#)
- npAddSquare\_8u32f\_C1IMR
  - image\_addsquare, [199](#)
- npAddSquare\_8u32f\_C1IR
  - image\_addsquare, [199](#)
- npAddWeighted\_16u32f\_C1IMR
  - image\_addweighted, [204](#)
- npAddWeighted\_16u32f\_C1IR
  - image\_addweighted, [205](#)
- npAddWeighted\_32f\_C1IMR
  - image\_addweighted, [205](#)
- npAddWeighted\_32f\_C1IR
  - image\_addweighted, [206](#)
- npAddWeighted\_8u32f\_C1IMR
  - image\_addweighted, [206](#)
- npAddWeighted\_8u32f\_C1IR
  - image\_addweighted, [206](#)
- npAlphaComp\_16s\_AC1R
  - image\_alphacomp, [489](#)
- npAlphaComp\_16u\_AC1R
  - image\_alphacomp, [489](#)
- npAlphaComp\_16u\_AC4R
  - image\_alphacomp, [490](#)
- npAlphaComp\_32f\_AC1R
  - image\_alphacomp, [490](#)
- npAlphaComp\_32f\_AC4R
  - image\_alphacomp, [491](#)
- npAlphaComp\_32s\_AC1R
  - image\_alphacomp, [491](#)
- npAlphaComp\_32s\_AC4R
  - image\_alphacomp, [491](#)
- npAlphaComp\_32u\_AC1R

- image\_alphacomp, [492](#)
- nppiAlphaComp\_32u\_AC4R
  - image\_alphacomp, [492](#)
- nppiAlphaComp\_8s\_AC1R
  - image\_alphacomp, [493](#)
- nppiAlphaComp\_8u\_AC1R
  - image\_alphacomp, [493](#)
- nppiAlphaComp\_8u\_AC4R
  - image\_alphacomp, [494](#)
- nppiAlphaCompC\_16s\_C1R
  - image\_alphacompc, [474](#)
- nppiAlphaCompC\_16u\_AC4R
  - image\_alphacompc, [474](#)
- nppiAlphaCompC\_16u\_C1R
  - image\_alphacompc, [475](#)
- nppiAlphaCompC\_16u\_C3R
  - image\_alphacompc, [475](#)
- nppiAlphaCompC\_16u\_C4R
  - image\_alphacompc, [476](#)
- nppiAlphaCompC\_32f\_C1R
  - image\_alphacompc, [476](#)
- nppiAlphaCompC\_32s\_C1R
  - image\_alphacompc, [477](#)
- nppiAlphaCompC\_32u\_C1R
  - image\_alphacompc, [477](#)
- nppiAlphaCompC\_8s\_C1R
  - image\_alphacompc, [478](#)
- nppiAlphaCompC\_8u\_AC4R
  - image\_alphacompc, [478](#)
- nppiAlphaCompC\_8u\_C1R
  - image\_alphacompc, [479](#)
- nppiAlphaCompC\_8u\_C3R
  - image\_alphacompc, [479](#)
- nppiAlphaCompC\_8u\_C4R
  - image\_alphacompc, [480](#)
- nppiAlphaCompColorKey\_8u\_AC4R
  - image\_complement\_color\_key, [612](#)
- NppiAlphaOp
  - typedefs\_npp, [41](#)
- nppiAlphaPremul\_16u\_AC4IR
  - image\_alphapremul, [495](#)
- nppiAlphaPremul\_16u\_AC4R
  - image\_alphapremul, [495](#)
- nppiAlphaPremul\_8u\_AC4IR
  - image\_alphapremul, [496](#)
- nppiAlphaPremul\_8u\_AC4R
  - image\_alphapremul, [496](#)
- nppiAlphaPremulC\_16u\_AC4IR
  - image\_alphapremulc, [482](#)
- nppiAlphaPremulC\_16u\_AC4R
  - image\_alphapremulc, [482](#)
- nppiAlphaPremulC\_16u\_C1IR
  - image\_alphapremulc, [483](#)
- nppiAlphaPremulC\_16u\_C1R
  - image\_alphapremulc, [483](#)
- nppiAlphaPremulC\_16u\_C3IR
  - image\_alphapremulc, [483](#)
- nppiAlphaPremulC\_16u\_C3R
  - image\_alphapremulc, [484](#)
- nppiAlphaPremulC\_16u\_C4IR
  - image\_alphapremulc, [484](#)
- nppiAlphaPremulC\_16u\_C4R
  - image\_alphapremulc, [484](#)
- nppiAlphaPremulC\_8u\_AC4IR
  - image\_alphapremulc, [485](#)
- nppiAlphaPremulC\_8u\_AC4R
  - image\_alphapremulc, [485](#)
- nppiAlphaPremulC\_8u\_C1IR
  - image\_alphapremulc, [485](#)
- nppiAlphaPremulC\_8u\_C1R
  - image\_alphapremulc, [486](#)
- nppiAlphaPremulC\_8u\_C3IR
  - image\_alphapremulc, [486](#)
- nppiAlphaPremulC\_8u\_C3R
  - image\_alphapremulc, [486](#)
- nppiAlphaPremulC\_8u\_C4IR
  - image\_alphapremulc, [487](#)
- nppiAlphaPremulC\_8u\_C4R
  - image\_alphapremulc, [487](#)
- nppiAnd\_16u\_AC4IR
  - image\_and, [434](#)
- nppiAnd\_16u\_AC4R
  - image\_and, [434](#)
- nppiAnd\_16u\_C1IR
  - image\_and, [434](#)
- nppiAnd\_16u\_C1R
  - image\_and, [435](#)
- nppiAnd\_16u\_C3IR
  - image\_and, [435](#)
- nppiAnd\_16u\_C3R
  - image\_and, [435](#)
- nppiAnd\_16u\_C4IR
  - image\_and, [436](#)
- nppiAnd\_16u\_C4R
  - image\_and, [436](#)
- nppiAnd\_32s\_AC4IR
  - image\_and, [437](#)
- nppiAnd\_32s\_AC4R
  - image\_and, [437](#)
- nppiAnd\_32s\_C1IR
  - image\_and, [437](#)
- nppiAnd\_32s\_C1R
  - image\_and, [438](#)
- nppiAnd\_32s\_C3IR
  - image\_and, [438](#)
- nppiAnd\_32s\_C3R
  - image\_and, [438](#)
- nppiAnd\_32s\_C4IR



- image\_and, [439](#)
- npPiAnd\_32s\_C4R
  - image\_and, [439](#)
- npPiAnd\_8u\_AC4IR
  - image\_and, [440](#)
- npPiAnd\_8u\_AC4R
  - image\_and, [440](#)
- npPiAnd\_8u\_C1IR
  - image\_and, [440](#)
- npPiAnd\_8u\_C1R
  - image\_and, [441](#)
- npPiAnd\_8u\_C3IR
  - image\_and, [441](#)
- npPiAnd\_8u\_C3R
  - image\_and, [441](#)
- npPiAnd\_8u\_C4IR
  - image\_and, [442](#)
- npPiAnd\_8u\_C4R
  - image\_and, [442](#)
- npPiAndC\_16u\_AC4IR
  - image\_andc, [373](#)
- npPiAndC\_16u\_AC4R
  - image\_andc, [373](#)
- npPiAndC\_16u\_C1IR
  - image\_andc, [373](#)
- npPiAndC\_16u\_C1R
  - image\_andc, [374](#)
- npPiAndC\_16u\_C3IR
  - image\_andc, [374](#)
- npPiAndC\_16u\_C3R
  - image\_andc, [374](#)
- npPiAndC\_16u\_C4IR
  - image\_andc, [375](#)
- npPiAndC\_16u\_C4R
  - image\_andc, [375](#)
- npPiAndC\_32s\_AC4IR
  - image\_andc, [375](#)
- npPiAndC\_32s\_AC4R
  - image\_andc, [376](#)
- npPiAndC\_32s\_C1IR
  - image\_andc, [376](#)
- npPiAndC\_32s\_C1R
  - image\_andc, [376](#)
- npPiAndC\_32s\_C3IR
  - image\_andc, [377](#)
- npPiAndC\_32s\_C3R
  - image\_andc, [377](#)
- npPiAndC\_32s\_C4IR
  - image\_andc, [377](#)
- npPiAndC\_32s\_C4R
  - image\_andc, [378](#)
- npPiAndC\_8u\_AC4IR
  - image\_andc, [378](#)
- npPiAndC\_8u\_AC4R
  - image\_andc, [378](#)
- image\_andc, [378](#)
- npPiAndC\_8u\_C1IR
  - image\_andc, [379](#)
- npPiAndC\_8u\_C1R
  - image\_andc, [379](#)
- npPiAndC\_8u\_C3IR
  - image\_andc, [379](#)
- npPiAndC\_8u\_C3R
  - image\_andc, [380](#)
- npPiAndC\_8u\_C4IR
  - image\_andc, [380](#)
- npPiAndC\_8u\_C4R
  - image\_andc, [380](#)
- npPiAverageError\_16s\_C1R
  - image\_average\_error, [2050](#)
- npPiAverageError\_16s\_C2R
  - image\_average\_error, [2051](#)
- npPiAverageError\_16s\_C3R
  - image\_average\_error, [2051](#)
- npPiAverageError\_16s\_C4R
  - image\_average\_error, [2052](#)
- npPiAverageError\_16sc\_C1R
  - image\_average\_error, [2052](#)
- npPiAverageError\_16sc\_C2R
  - image\_average\_error, [2052](#)
- npPiAverageError\_16sc\_C3R
  - image\_average\_error, [2053](#)
- npPiAverageError\_16sc\_C4R
  - image\_average\_error, [2053](#)
- npPiAverageError\_16u\_C1R
  - image\_average\_error, [2054](#)
- npPiAverageError\_16u\_C2R
  - image\_average\_error, [2054](#)
- npPiAverageError\_16u\_C3R
  - image\_average\_error, [2055](#)
- npPiAverageError\_16u\_C4R
  - image\_average\_error, [2055](#)
- npPiAverageError\_32f\_C1R
  - image\_average\_error, [2055](#)
- npPiAverageError\_32f\_C2R
  - image\_average\_error, [2056](#)
- npPiAverageError\_32f\_C3R
  - image\_average\_error, [2056](#)
- npPiAverageError\_32f\_C4R
  - image\_average\_error, [2057](#)
- npPiAverageError\_32fc\_C1R
  - image\_average\_error, [2057](#)
- npPiAverageError\_32fc\_C2R
  - image\_average\_error, [2058](#)
- npPiAverageError\_32fc\_C3R
  - image\_average\_error, [2058](#)
- npPiAverageError\_32fc\_C4R
  - image\_average\_error, [2059](#)
- npPiAverageError\_32s\_C1R



- image\_average\_error, 2059
- nppiAverageError\_32s\_C2R
  - image\_average\_error, 2059
- nppiAverageError\_32s\_C3R
  - image\_average\_error, 2060
- nppiAverageError\_32s\_C4R
  - image\_average\_error, 2060
- nppiAverageError\_32sc\_C1R
  - image\_average\_error, 2061
- nppiAverageError\_32sc\_C2R
  - image\_average\_error, 2061
- nppiAverageError\_32sc\_C3R
  - image\_average\_error, 2062
- nppiAverageError\_32sc\_C4R
  - image\_average\_error, 2062
- nppiAverageError\_32u\_C1R
  - image\_average\_error, 2062
- nppiAverageError\_32u\_C2R
  - image\_average\_error, 2063
- nppiAverageError\_32u\_C3R
  - image\_average\_error, 2063
- nppiAverageError\_32u\_C4R
  - image\_average\_error, 2064
- nppiAverageError\_64f\_C1R
  - image\_average\_error, 2064
- nppiAverageError\_64f\_C2R
  - image\_average\_error, 2065
- nppiAverageError\_64f\_C3R
  - image\_average\_error, 2065
- nppiAverageError\_64f\_C4R
  - image\_average\_error, 2066
- nppiAverageError\_8s\_C1R
  - image\_average\_error, 2066
- nppiAverageError\_8s\_C2R
  - image\_average\_error, 2066
- nppiAverageError\_8s\_C3R
  - image\_average\_error, 2067
- nppiAverageError\_8s\_C4R
  - image\_average\_error, 2067
- nppiAverageError\_8u\_C1R
  - image\_average\_error, 2068
- nppiAverageError\_8u\_C2R
  - image\_average\_error, 2068
- nppiAverageError\_8u\_C3R
  - image\_average\_error, 2069
- nppiAverageError\_8u\_C4R
  - image\_average\_error, 2069
- nppiAverageErrorGetBufferHostSize\_16s\_C1R
  - image\_statistics\_functions, 1409
- nppiAverageErrorGetBufferHostSize\_16s\_C2R
  - image\_statistics\_functions, 1409
- nppiAverageErrorGetBufferHostSize\_16s\_C3R
  - image\_statistics\_functions, 1409
- nppiAverageErrorGetBufferHostSize\_16s\_C4R
  - image\_statistics\_functions, 1409
- nppiAverageErrorGetBufferHostSize\_16sc\_C1R
  - image\_statistics\_functions, 1410
- nppiAverageErrorGetBufferHostSize\_16sc\_C2R
  - image\_statistics\_functions, 1410
- nppiAverageErrorGetBufferHostSize\_16sc\_C3R
  - image\_statistics\_functions, 1410
- nppiAverageErrorGetBufferHostSize\_16sc\_C4R
  - image\_statistics\_functions, 1411
- nppiAverageErrorGetBufferHostSize\_16u\_C1R
  - image\_statistics\_functions, 1411
- nppiAverageErrorGetBufferHostSize\_16u\_C2R
  - image\_statistics\_functions, 1411
- nppiAverageErrorGetBufferHostSize\_16u\_C3R
  - image\_statistics\_functions, 1411
- nppiAverageErrorGetBufferHostSize\_16u\_C4R
  - image\_statistics\_functions, 1412
- nppiAverageErrorGetBufferHostSize\_32f\_C1R
  - image\_statistics\_functions, 1412
- nppiAverageErrorGetBufferHostSize\_32f\_C2R
  - image\_statistics\_functions, 1412
- nppiAverageErrorGetBufferHostSize\_32f\_C3R
  - image\_statistics\_functions, 1413
- nppiAverageErrorGetBufferHostSize\_32f\_C4R
  - image\_statistics\_functions, 1413
- nppiAverageErrorGetBufferHostSize\_32fc\_C1R
  - image\_statistics\_functions, 1413
- nppiAverageErrorGetBufferHostSize\_32fc\_C2R
  - image\_statistics\_functions, 1413
- nppiAverageErrorGetBufferHostSize\_32fc\_C3R
  - image\_statistics\_functions, 1414
- nppiAverageErrorGetBufferHostSize\_32fc\_C4R
  - image\_statistics\_functions, 1414
- nppiAverageErrorGetBufferHostSize\_32s\_C1R
  - image\_statistics\_functions, 1414
- nppiAverageErrorGetBufferHostSize\_32s\_C2R
  - image\_statistics\_functions, 1415
- nppiAverageErrorGetBufferHostSize\_32s\_C3R
  - image\_statistics\_functions, 1415
- nppiAverageErrorGetBufferHostSize\_32s\_C4R
  - image\_statistics\_functions, 1415
- nppiAverageErrorGetBufferHostSize\_32sc\_C1R
  - image\_statistics\_functions, 1415
- nppiAverageErrorGetBufferHostSize\_32sc\_C2R
  - image\_statistics\_functions, 1416
- nppiAverageErrorGetBufferHostSize\_32sc\_C3R
  - image\_statistics\_functions, 1416
- nppiAverageErrorGetBufferHostSize\_32sc\_C4R
  - image\_statistics\_functions, 1416
- nppiAverageErrorGetBufferHostSize\_32u\_C1R
  - image\_statistics\_functions, 1417
- nppiAverageErrorGetBufferHostSize\_32u\_C2R
  - image\_statistics\_functions, 1417
- nppiAverageErrorGetBufferHostSize\_32u\_C3R
  - image\_statistics\_functions, 1417

image\_statistics\_functions, [1417](#)  
 nppiAverageErrorGetBufferHostSize\_32u\_C4R  
     image\_statistics\_functions, [1417](#)  
 nppiAverageErrorGetBufferHostSize\_64f\_C1R  
     image\_statistics\_functions, [1418](#)  
 nppiAverageErrorGetBufferHostSize\_64f\_C2R  
     image\_statistics\_functions, [1418](#)  
 nppiAverageErrorGetBufferHostSize\_64f\_C3R  
     image\_statistics\_functions, [1418](#)  
 nppiAverageErrorGetBufferHostSize\_64f\_C4R  
     image\_statistics\_functions, [1419](#)  
 nppiAverageErrorGetBufferHostSize\_8s\_C1R  
     image\_statistics\_functions, [1419](#)  
 nppiAverageErrorGetBufferHostSize\_8s\_C2R  
     image\_statistics\_functions, [1419](#)  
 nppiAverageErrorGetBufferHostSize\_8s\_C3R  
     image\_statistics\_functions, [1419](#)  
 nppiAverageErrorGetBufferHostSize\_8s\_C4R  
     image\_statistics\_functions, [1420](#)  
 nppiAverageErrorGetBufferHostSize\_8u\_C1R  
     image\_statistics\_functions, [1420](#)  
 nppiAverageErrorGetBufferHostSize\_8u\_C2R  
     image\_statistics\_functions, [1420](#)  
 nppiAverageErrorGetBufferHostSize\_8u\_C3R  
     image\_statistics\_functions, [1421](#)  
 nppiAverageErrorGetBufferHostSize\_8u\_C4R  
     image\_statistics\_functions, [1421](#)  
 nppiAverageRelativeError\_16s\_C1R  
     image\_average\_relative\_error, [2097](#)  
 nppiAverageRelativeError\_16s\_C2R  
     image\_average\_relative\_error, [2098](#)  
 nppiAverageRelativeError\_16s\_C3R  
     image\_average\_relative\_error, [2098](#)  
 nppiAverageRelativeError\_16s\_C4R  
     image\_average\_relative\_error, [2099](#)  
 nppiAverageRelativeError\_16sc\_C1R  
     image\_average\_relative\_error, [2099](#)  
 nppiAverageRelativeError\_16sc\_C2R  
     image\_average\_relative\_error, [2100](#)  
 nppiAverageRelativeError\_16sc\_C3R  
     image\_average\_relative\_error, [2100](#)  
 nppiAverageRelativeError\_16sc\_C4R  
     image\_average\_relative\_error, [2100](#)  
 nppiAverageRelativeError\_16u\_C1R  
     image\_average\_relative\_error, [2101](#)  
 nppiAverageRelativeError\_16u\_C2R  
     image\_average\_relative\_error, [2101](#)  
 nppiAverageRelativeError\_16u\_C3R  
     image\_average\_relative\_error, [2102](#)  
 nppiAverageRelativeError\_16u\_C4R  
     image\_average\_relative\_error, [2102](#)  
 nppiAverageRelativeError\_32f\_C1R  
     image\_average\_relative\_error, [2103](#)  
 nppiAverageRelativeError\_32f\_C2R

image\_average\_relative\_error, [2103](#)  
 nppiAverageRelativeError\_32f\_C3R  
     image\_average\_relative\_error, [2104](#)  
 nppiAverageRelativeError\_32f\_C4R  
     image\_average\_relative\_error, [2104](#)  
 nppiAverageRelativeError\_32fc\_C1R  
     image\_average\_relative\_error, [2105](#)  
 nppiAverageRelativeError\_32fc\_C2R  
     image\_average\_relative\_error, [2105](#)  
 nppiAverageRelativeError\_32fc\_C3R  
     image\_average\_relative\_error, [2105](#)  
 nppiAverageRelativeError\_32fc\_C4R  
     image\_average\_relative\_error, [2106](#)  
 nppiAverageRelativeError\_32s\_C1R  
     image\_average\_relative\_error, [2106](#)  
 nppiAverageRelativeError\_32s\_C2R  
     image\_average\_relative\_error, [2107](#)  
 nppiAverageRelativeError\_32s\_C3R  
     image\_average\_relative\_error, [2107](#)  
 nppiAverageRelativeError\_32s\_C4R  
     image\_average\_relative\_error, [2108](#)  
 nppiAverageRelativeError\_32sc\_C1R  
     image\_average\_relative\_error, [2108](#)  
 nppiAverageRelativeError\_32sc\_C2R  
     image\_average\_relative\_error, [2109](#)  
 nppiAverageRelativeError\_32sc\_C3R  
     image\_average\_relative\_error, [2109](#)  
 nppiAverageRelativeError\_32sc\_C4R  
     image\_average\_relative\_error, [2110](#)  
 nppiAverageRelativeError\_32u\_C1R  
     image\_average\_relative\_error, [2110](#)  
 nppiAverageRelativeError\_32u\_C2R  
     image\_average\_relative\_error, [2110](#)  
 nppiAverageRelativeError\_32u\_C3R  
     image\_average\_relative\_error, [2111](#)  
 nppiAverageRelativeError\_32u\_C4R  
     image\_average\_relative\_error, [2111](#)  
 nppiAverageRelativeError\_64f\_C1R  
     image\_average\_relative\_error, [2112](#)  
 nppiAverageRelativeError\_64f\_C2R  
     image\_average\_relative\_error, [2112](#)  
 nppiAverageRelativeError\_64f\_C3R  
     image\_average\_relative\_error, [2113](#)  
 nppiAverageRelativeError\_64f\_C4R  
     image\_average\_relative\_error, [2113](#)  
 nppiAverageRelativeError\_8s\_C1R  
     image\_average\_relative\_error, [2114](#)  
 nppiAverageRelativeError\_8s\_C2R  
     image\_average\_relative\_error, [2114](#)  
 nppiAverageRelativeError\_8s\_C3R  
     image\_average\_relative\_error, [2115](#)  
 nppiAverageRelativeError\_8s\_C4R  
     image\_average\_relative\_error, [2115](#)  
 nppiAverageRelativeError\_8u\_C1R

- image\_average\_relative\_error, [2115](#)
- nppiAverageRelativeError\_8u\_C2R
  - image\_average\_relative\_error, [2116](#)
- nppiAverageRelativeError\_8u\_C3R
  - image\_average\_relative\_error, [2116](#)
- nppiAverageRelativeError\_8u\_C4R
  - image\_average\_relative\_error, [2117](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16s\_C1R
  - image\_statistics\_functions, [1421](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16s\_C2R
  - image\_statistics\_functions, [1421](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16s\_C3R
  - image\_statistics\_functions, [1422](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16s\_C4R
  - image\_statistics\_functions, [1422](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16sc\_C1R
  - image\_statistics\_functions, [1422](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16sc\_C2R
  - image\_statistics\_functions, [1423](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16sc\_C3R
  - image\_statistics\_functions, [1423](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16sc\_C4R
  - image\_statistics\_functions, [1423](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16u\_C1R
  - image\_statistics\_functions, [1423](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16u\_C2R
  - image\_statistics\_functions, [1424](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16u\_C3R
  - image\_statistics\_functions, [1424](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-16u\_C4R
  - image\_statistics\_functions, [1424](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32f\_C1R
  - image\_statistics\_functions, [1425](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32f\_C2R
  - image\_statistics\_functions, [1425](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32f\_C3R
  - image\_statistics\_functions, [1425](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32f\_C4R
  - image\_statistics\_functions, [1425](#)
- image\_statistics\_functions, [1425](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32fc\_C1R
  - image\_statistics\_functions, [1426](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32fc\_C2R
  - image\_statistics\_functions, [1426](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32fc\_C3R
  - image\_statistics\_functions, [1426](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32fc\_C4R
  - image\_statistics\_functions, [1427](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32s\_C1R
  - image\_statistics\_functions, [1427](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32s\_C2R
  - image\_statistics\_functions, [1427](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32s\_C3R
  - image\_statistics\_functions, [1427](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32s\_C4R
  - image\_statistics\_functions, [1428](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32sc\_C1R
  - image\_statistics\_functions, [1428](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32sc\_C2R
  - image\_statistics\_functions, [1428](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32sc\_C3R
  - image\_statistics\_functions, [1429](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32sc\_C4R
  - image\_statistics\_functions, [1429](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32u\_C1R
  - image\_statistics\_functions, [1429](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32u\_C2R
  - image\_statistics\_functions, [1429](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32u\_C3R
  - image\_statistics\_functions, [1430](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-32u\_C4R
  - image\_statistics\_functions, [1430](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-64f\_C1R
  - image\_statistics\_functions, [1430](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-64f\_C2R
  - image\_statistics\_functions, [1430](#)

- image\_statistics\_functions, [1431](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-64f\_C3R
  - image\_statistics\_functions, [1431](#)
- nppiAverageRelativeErrorGetBufferHostSize\_-64f\_C4R
  - image\_statistics\_functions, [1431](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8s\_-C1R
  - image\_statistics\_functions, [1431](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8s\_-C2R
  - image\_statistics\_functions, [1432](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8s\_-C3R
  - image\_statistics\_functions, [1432](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8s\_-C4R
  - image\_statistics\_functions, [1432](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8u\_-C1R
  - image\_statistics\_functions, [1433](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8u\_-C2R
  - image\_statistics\_functions, [1433](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8u\_-C3R
  - image\_statistics\_functions, [1433](#)
- nppiAverageRelativeErrorGetBufferHostSize\_8u\_-C4R
  - image\_statistics\_functions, [1433](#)
- NppiAxis
  - typedefs\_npp, [42](#)
- nppiBGRToCbYCr422\_709HDTV\_8u\_AC4C2R
  - image\_color\_model\_conversion, [525](#)
- nppiBGRToCbYCr422\_709HDTV\_8u\_C3C2R
  - image\_color\_model\_conversion, [525](#)
- nppiBGRToCbYCr422\_8u\_AC4C2R
  - image\_color\_model\_conversion, [525](#)
- nppiBGRToHLS\_8u\_AC4P4R
  - image\_color\_model\_conversion, [526](#)
- nppiBGRToHLS\_8u\_AC4R
  - image\_color\_model\_conversion, [526](#)
- nppiBGRToHLS\_8u\_AP4C4R
  - image\_color\_model\_conversion, [526](#)
- nppiBGRToHLS\_8u\_AP4R
  - image\_color\_model\_conversion, [527](#)
- nppiBGRToHLS\_8u\_C3P3R
  - image\_color\_model\_conversion, [527](#)
- nppiBGRToHLS\_8u\_P3C3R
  - image\_color\_model\_conversion, [527](#)
- nppiBGRToHLS\_8u\_P3R
  - image\_color\_model\_conversion, [528](#)
- nppiBGRToLab\_8u\_C3R
  - image\_color\_model\_conversion, [528](#)
- nppiBGRToYCbCr411\_8u\_AC4P3R
  - image\_color\_model\_conversion, [528](#)
- nppiBGRToYCbCr411\_8u\_C3P3R
  - image\_color\_model\_conversion, [529](#)
- nppiBGRToYCbCr420\_709CSC\_8u\_AC4P3R
  - image\_color\_model\_conversion, [529](#)
- nppiBGRToYCbCr420\_709CSC\_8u\_C3P3R
  - image\_color\_model\_conversion, [530](#)
- nppiBGRToYCbCr420\_709HDTV\_8u\_AC4P3R
  - image\_color\_model\_conversion, [530](#)
- nppiBGRToYCbCr420\_8u\_AC4P3R
  - image\_color\_model\_conversion, [530](#)
- nppiBGRToYCbCr420\_8u\_C3P3R
  - image\_color\_model\_conversion, [531](#)
- nppiBGRToYCbCr422\_8u\_AC4C2R
  - image\_color\_model\_conversion, [531](#)
- nppiBGRToYCbCr422\_8u\_AC4P3R
  - image\_color\_model\_conversion, [532](#)
- nppiBGRToYCbCr422\_8u\_C3C2R
  - image\_color\_model\_conversion, [532](#)
- nppiBGRToYCbCr422\_8u\_C3P3R
  - image\_color\_model\_conversion, [532](#)
- nppiBGRToYCbCr\_8u\_AC4P3R
  - image\_color\_model\_conversion, [533](#)
- nppiBGRToYCbCr\_8u\_AC4P4R
  - image\_color\_model\_conversion, [533](#)
- nppiBGRToYCbCr\_8u\_C3P3R
  - image\_color\_model\_conversion, [534](#)
- nppiBGRToYCrCb420\_709CSC\_8u\_AC4P3R
  - image\_color\_model\_conversion, [534](#)
- nppiBGRToYCrCb420\_709CSC\_8u\_C3P3R
  - image\_color\_model\_conversion, [534](#)
- nppiBGRToYCrCb420\_8u\_AC4P3R
  - image\_color\_model\_conversion, [535](#)
- nppiBGRToYCrCb420\_8u\_C3P3R
  - image\_color\_model\_conversion, [535](#)
- nppiBGRToYUV420\_8u\_AC4P3R
  - image\_color\_model\_conversion, [536](#)
- nppiBGRToYUV\_8u\_AC4P4R
  - image\_color\_model\_conversion, [536](#)
- nppiBGRToYUV\_8u\_AC4R
  - image\_color\_model\_conversion, [536](#)
- nppiBGRToYUV\_8u\_C3P3R
  - image\_color\_model\_conversion, [537](#)
- nppiBGRToYUV\_8u\_C3R
  - image\_color\_model\_conversion, [537](#)
- nppiBGRToYUV\_8u\_P3R
  - image\_color\_model\_conversion, [537](#)
- NppiBorderType
  - typedefs\_npp, [42](#)
- nppiCbYCr422ToBGR\_709HDTV\_8u\_C2C3R
  - image\_color\_model\_conversion, [538](#)
- nppiCbYCr422ToBGR\_709HDTV\_8u\_C2C4R

- image\_color\_model\_conversion, 538
- nppiCbYCr422ToBGR\_8u\_C2C4R
  - image\_color\_model\_conversion, 539
- nppiCbYCr422ToRGB\_8u\_C2C3R
  - image\_color\_model\_conversion, 539
- nppiCbYCr422ToYCbCr411\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, 585
- nppiCbYCr422ToYCbCr420\_8u\_C2P2R
  - image\_color\_sampling\_format\_conversion, 586
- nppiCbYCr422ToYCbCr420\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, 586
- nppiCbYCr422ToYCbCr422\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, 586
- nppiCbYCr422ToYCbCr422\_8u\_C2R
  - image\_color\_sampling\_format\_conversion, 587
- nppiCbYCr422ToYCrCb420\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, 587
- nppiColorToGray\_16s\_AC4C1R
  - image\_color\_model\_conversion, 539
- nppiColorToGray\_16s\_C3C1R
  - image\_color\_model\_conversion, 540
- nppiColorToGray\_16u\_AC4C1R
  - image\_color\_model\_conversion, 540
- nppiColorToGray\_16u\_C3C1R
  - image\_color\_model\_conversion, 540
- nppiColorToGray\_32f\_AC4C1R
  - image\_color\_model\_conversion, 541
- nppiColorToGray\_32f\_C3C1R
  - image\_color\_model\_conversion, 541
- nppiColorToGray\_8u\_AC4C1R
  - image\_color\_model\_conversion, 542
- nppiColorToGray\_8u\_C3C1R
  - image\_color\_model\_conversion, 542
- nppiColorTwist32f\_16s\_AC4IR
  - image\_color\_processing, 628
- nppiColorTwist32f\_16s\_AC4R
  - image\_color\_processing, 628
- nppiColorTwist32f\_16s\_C3IR
  - image\_color\_processing, 628
- nppiColorTwist32f\_16s\_C3R
  - image\_color\_processing, 629
- nppiColorTwist32f\_16s\_IP3R
  - image\_color\_processing, 629
- nppiColorTwist32f\_16s\_P3R
  - image\_color\_processing, 629
- nppiColorTwist32f\_16u\_AC4IR
  - image\_color\_processing, 630
- nppiColorTwist32f\_16u\_AC4R
  - image\_color\_processing, 630
- nppiColorTwist32f\_16u\_C3IR
  - image\_color\_processing, 631
- nppiColorTwist32f\_16u\_C3R
  - image\_color\_processing, 631
- nppiColorTwist32f\_16u\_IP3R
  - image\_color\_processing, 631
- nppiColorTwist32f\_16u\_P3R
  - image\_color\_processing, 632
- nppiColorTwist32f\_8s\_AC4IR
  - image\_color\_processing, 632
- nppiColorTwist32f\_8s\_AC4R
  - image\_color\_processing, 633
- nppiColorTwist32f\_8s\_C3IR
  - image\_color\_processing, 633
- nppiColorTwist32f\_8s\_C3R
  - image\_color\_processing, 633
- nppiColorTwist32f\_8s\_C4IR
  - image\_color\_processing, 634
- nppiColorTwist32f\_8s\_C4R
  - image\_color\_processing, 634
- nppiColorTwist32f\_8s\_IP3R
  - image\_color\_processing, 634
- nppiColorTwist32f\_8s\_P3R
  - image\_color\_processing, 635
- nppiColorTwist32f\_8u\_AC4IR
  - image\_color\_processing, 635
- nppiColorTwist32f\_8u\_AC4R
  - image\_color\_processing, 636
- nppiColorTwist32f\_8u\_C3IR
  - image\_color\_processing, 636
- nppiColorTwist32f\_8u\_C3R
  - image\_color\_processing, 636
- nppiColorTwist32f\_8u\_C4IR
  - image\_color\_processing, 637
- nppiColorTwist32f\_8u\_C4R
  - image\_color\_processing, 637
- nppiColorTwist32f\_8u\_IP3R
  - image\_color\_processing, 637
- nppiColorTwist32f\_8u\_P3R
  - image\_color\_processing, 638
- nppiColorTwist\_32f\_AC4IR
  - image\_color\_processing, 638
- nppiColorTwist\_32f\_AC4R
  - image\_color\_processing, 639
- nppiColorTwist\_32f\_C3IR
  - image\_color\_processing, 639
- nppiColorTwist\_32f\_C3R
  - image\_color\_processing, 639
- nppiColorTwist\_32f\_C4IR
  - image\_color\_processing, 640
- nppiColorTwist\_32f\_C4R
  - image\_color\_processing, 640
- nppiColorTwist\_32f\_IP3R



- image\_color\_processing, 640
- npPiColorTwist\_32f\_P3R
  - image\_color\_processing, 641
- npPiCompare\_16s\_AC4R
  - image\_compare\_operations, 2223
- npPiCompare\_16s\_C1R
  - image\_compare\_operations, 2224
- npPiCompare\_16s\_C3R
  - image\_compare\_operations, 2224
- npPiCompare\_16s\_C4R
  - image\_compare\_operations, 2224
- npPiCompare\_16u\_AC4R
  - image\_compare\_operations, 2225
- npPiCompare\_16u\_C1R
  - image\_compare\_operations, 2225
- npPiCompare\_16u\_C3R
  - image\_compare\_operations, 2226
- npPiCompare\_16u\_C4R
  - image\_compare\_operations, 2226
- npPiCompare\_32f\_AC4R
  - image\_compare\_operations, 2227
- npPiCompare\_32f\_C1R
  - image\_compare\_operations, 2227
- npPiCompare\_32f\_C3R
  - image\_compare\_operations, 2228
- npPiCompare\_32f\_C4R
  - image\_compare\_operations, 2228
- npPiCompare\_8u\_AC4R
  - image\_compare\_operations, 2229
- npPiCompare\_8u\_C1R
  - image\_compare\_operations, 2229
- npPiCompare\_8u\_C3R
  - image\_compare\_operations, 2230
- npPiCompare\_8u\_C4R
  - image\_compare\_operations, 2230
- npPiCompareC\_16s\_AC4R
  - image\_compare\_operations, 2231
- npPiCompareC\_16s\_C1R
  - image\_compare\_operations, 2231
- npPiCompareC\_16s\_C3R
  - image\_compare\_operations, 2232
- npPiCompareC\_16s\_C4R
  - image\_compare\_operations, 2232
- npPiCompareC\_16u\_AC4R
  - image\_compare\_operations, 2233
- npPiCompareC\_16u\_C1R
  - image\_compare\_operations, 2233
- npPiCompareC\_16u\_C3R
  - image\_compare\_operations, 2233
- npPiCompareC\_16u\_C4R
  - image\_compare\_operations, 2234
- npPiCompareC\_32f\_AC4R
  - image\_compare\_operations, 2234
- npPiCompareC\_32f\_C1R
  - image\_compare\_operations, 2235
- npPiCompareC\_32f\_C3R
  - image\_compare\_operations, 2235
- npPiCompareC\_32f\_C4R
  - image\_compare\_operations, 2236
- npPiCompareC\_8u\_AC4R
  - image\_compare\_operations, 2236
- npPiCompareC\_8u\_C1R
  - image\_compare\_operations, 2236
- npPiCompareC\_8u\_C3R
  - image\_compare\_operations, 2237
- npPiCompareC\_8u\_C4R
  - image\_compare\_operations, 2237
- npPiCompareEqualEps\_32f\_AC4R
  - image\_compare\_operations, 2238
- npPiCompareEqualEps\_32f\_C1R
  - image\_compare\_operations, 2238
- npPiCompareEqualEps\_32f\_C3R
  - image\_compare\_operations, 2239
- npPiCompareEqualEps\_32f\_C4R
  - image\_compare\_operations, 2239
- npPiCompareEqualEpsC\_32f\_AC4R
  - image\_compare\_operations, 2240
- npPiCompareEqualEpsC\_32f\_C1R
  - image\_compare\_operations, 2240
- npPiCompareEqualEpsC\_32f\_C3R
  - image\_compare\_operations, 2241
- npPiCompareEqualEpsC\_32f\_C4R
  - image\_compare\_operations, 2241
- npPiCompColorKey\_8u\_C1R
  - image\_complement\_color\_key, 613
- npPiCompColorKey\_8u\_C3R
  - image\_complement\_color\_key, 613
- npPiCompColorKey\_8u\_C4R
  - image\_complement\_color\_key, 614
- npPiConvert\_16s16u\_C1Rs
  - image\_convert, 809
- npPiConvert\_16s32f\_AC4R
  - image\_convert, 809
- npPiConvert\_16s32f\_C1R
  - image\_convert, 810
- npPiConvert\_16s32f\_C3R
  - image\_convert, 810
- npPiConvert\_16s32f\_C4R
  - image\_convert, 810
- npPiConvert\_16s32s\_AC4R
  - image\_convert, 811
- npPiConvert\_16s32s\_C1R
  - image\_convert, 811
- npPiConvert\_16s32s\_C3R
  - image\_convert, 811
- npPiConvert\_16s32s\_C4R
  - image\_convert, 812
- npPiConvert\_16s32u\_C1Rs

- image\_convert, [812](#)
- nppiConvert\_16s8s\_C1RSfs
  - image\_convert, [812](#)
- nppiConvert\_16s8u\_AC4R
  - image\_convert, [813](#)
- nppiConvert\_16s8u\_C1R
  - image\_convert, [813](#)
- nppiConvert\_16s8u\_C3R
  - image\_convert, [813](#)
- nppiConvert\_16s8u\_C4R
  - image\_convert, [814](#)
- nppiConvert\_16u16s\_C1RSfs
  - image\_convert, [814](#)
- nppiConvert\_16u32f\_AC4R
  - image\_convert, [814](#)
- nppiConvert\_16u32f\_C1R
  - image\_convert, [815](#)
- nppiConvert\_16u32f\_C3R
  - image\_convert, [815](#)
- nppiConvert\_16u32f\_C4R
  - image\_convert, [815](#)
- nppiConvert\_16u32s\_AC4R
  - image\_convert, [816](#)
- nppiConvert\_16u32s\_C1R
  - image\_convert, [816](#)
- nppiConvert\_16u32s\_C3R
  - image\_convert, [816](#)
- nppiConvert\_16u32s\_C4R
  - image\_convert, [817](#)
- nppiConvert\_16u32u\_C1R
  - image\_convert, [817](#)
- nppiConvert\_16u8s\_C1RSfs
  - image\_convert, [817](#)
- nppiConvert\_16u8u\_AC4R
  - image\_convert, [818](#)
- nppiConvert\_16u8u\_C1R
  - image\_convert, [818](#)
- nppiConvert\_16u8u\_C3R
  - image\_convert, [818](#)
- nppiConvert\_16u8u\_C4R
  - image\_convert, [819](#)
- nppiConvert\_32f16s\_AC4R
  - image\_convert, [819](#)
- nppiConvert\_32f16s\_C1R
  - image\_convert, [819](#)
- nppiConvert\_32f16s\_C1RSfs
  - image\_convert, [820](#)
- nppiConvert\_32f16s\_C3R
  - image\_convert, [820](#)
- nppiConvert\_32f16s\_C4R
  - image\_convert, [821](#)
- nppiConvert\_32f16u\_AC4R
  - image\_convert, [821](#)
- nppiConvert\_32f16u\_C1R
  - image\_convert, [821](#)
- nppiConvert\_32f16u\_C1RSfs
  - image\_convert, [822](#)
- nppiConvert\_32f16u\_C3R
  - image\_convert, [822](#)
- nppiConvert\_32f16u\_C4R
  - image\_convert, [823](#)
- nppiConvert\_32f32s\_C1RSfs
  - image\_convert, [823](#)
- nppiConvert\_32f32u\_C1RSfs
  - image\_convert, [823](#)
- nppiConvert\_32f8s\_AC4R
  - image\_convert, [824](#)
- nppiConvert\_32f8s\_C1R
  - image\_convert, [824](#)
- nppiConvert\_32f8s\_C1RSfs
  - image\_convert, [825](#)
- nppiConvert\_32f8s\_C3R
  - image\_convert, [825](#)
- nppiConvert\_32f8s\_C4R
  - image\_convert, [825](#)
- nppiConvert\_32f8u\_AC4R
  - image\_convert, [826](#)
- nppiConvert\_32f8u\_C1R
  - image\_convert, [826](#)
- nppiConvert\_32f8u\_C1RSfs
  - image\_convert, [826](#)
- nppiConvert\_32f8u\_C3R
  - image\_convert, [827](#)
- nppiConvert\_32f8u\_C4R
  - image\_convert, [827](#)
- nppiConvert\_32s16s\_C1RSfs
  - image\_convert, [828](#)
- nppiConvert\_32s16u\_C1RSfs
  - image\_convert, [828](#)
- nppiConvert\_32s32f\_C1R
  - image\_convert, [828](#)
- nppiConvert\_32s32u\_C1Rs
  - image\_convert, [829](#)
- nppiConvert\_32s8s\_AC4R
  - image\_convert, [829](#)
- nppiConvert\_32s8s\_C1R
  - image\_convert, [829](#)
- nppiConvert\_32s8s\_C3R
  - image\_convert, [830](#)
- nppiConvert\_32s8s\_C4R
  - image\_convert, [830](#)
- nppiConvert\_32s8u\_AC4R
  - image\_convert, [830](#)
- nppiConvert\_32s8u\_C1R
  - image\_convert, [831](#)
- nppiConvert\_32s8u\_C3R
  - image\_convert, [831](#)
- nppiConvert\_32s8u\_C4R

image\_convert, [831](#)  
 nppiConvert\_32u16s\_C1RSfs  
   image\_convert, [832](#)  
 nppiConvert\_32u16u\_C1RSfs  
   image\_convert, [832](#)  
 nppiConvert\_32u32f\_C1R  
   image\_convert, [833](#)  
 nppiConvert\_32u32s\_C1RSfs  
   image\_convert, [833](#)  
 nppiConvert\_32u8s\_C1RSfs  
   image\_convert, [833](#)  
 nppiConvert\_32u8u\_C1RSfs  
   image\_convert, [834](#)  
 nppiConvert\_8s16s\_C1R  
   image\_convert, [834](#)  
 nppiConvert\_8s16u\_C1Rs  
   image\_convert, [835](#)  
 nppiConvert\_8s32f\_AC4R  
   image\_convert, [835](#)  
 nppiConvert\_8s32f\_C1R  
   image\_convert, [835](#)  
 nppiConvert\_8s32f\_C3R  
   image\_convert, [836](#)  
 nppiConvert\_8s32f\_C4R  
   image\_convert, [836](#)  
 nppiConvert\_8s32s\_AC4R  
   image\_convert, [836](#)  
 nppiConvert\_8s32s\_C1R  
   image\_convert, [837](#)  
 nppiConvert\_8s32s\_C3R  
   image\_convert, [837](#)  
 nppiConvert\_8s32s\_C4R  
   image\_convert, [837](#)  
 nppiConvert\_8s32u\_C1Rs  
   image\_convert, [838](#)  
 nppiConvert\_8s8u\_C1Rs  
   image\_convert, [838](#)  
 nppiConvert\_8u16s\_AC4R  
   image\_convert, [838](#)  
 nppiConvert\_8u16s\_C1R  
   image\_convert, [839](#)  
 nppiConvert\_8u16s\_C3R  
   image\_convert, [839](#)  
 nppiConvert\_8u16s\_C4R  
   image\_convert, [839](#)  
 nppiConvert\_8u16u\_AC4R  
   image\_convert, [840](#)  
 nppiConvert\_8u16u\_C1R  
   image\_convert, [840](#)  
 nppiConvert\_8u16u\_C3R  
   image\_convert, [840](#)  
 nppiConvert\_8u16u\_C4R  
   image\_convert, [841](#)  
 nppiConvert\_8u32f\_AC4R

image\_convert, [841](#)  
 nppiConvert\_8u32f\_C1R  
   image\_convert, [841](#)  
 nppiConvert\_8u32f\_C3R  
   image\_convert, [842](#)  
 nppiConvert\_8u32f\_C4R  
   image\_convert, [842](#)  
 nppiConvert\_8u32s\_AC4R  
   image\_convert, [842](#)  
 nppiConvert\_8u32s\_C1R  
   image\_convert, [843](#)  
 nppiConvert\_8u32s\_C3R  
   image\_convert, [843](#)  
 nppiConvert\_8u32s\_C4R  
   image\_convert, [843](#)  
 nppiConvert\_8u8s\_C1RSfs  
   image\_convert, [844](#)  
 nppiCopy\_16s\_AC4MR  
   image\_copy, [763](#)  
 nppiCopy\_16s\_AC4R  
   image\_copy, [764](#)  
 nppiCopy\_16s\_C1C3R  
   image\_copy, [764](#)  
 nppiCopy\_16s\_C1C4R  
   image\_copy, [764](#)  
 nppiCopy\_16s\_C1MR  
   image\_copy, [765](#)  
 nppiCopy\_16s\_C1R  
   image\_copy, [765](#)  
 nppiCopy\_16s\_C3C1R  
   image\_copy, [765](#)  
 nppiCopy\_16s\_C3CR  
   image\_copy, [766](#)  
 nppiCopy\_16s\_C3MR  
   image\_copy, [766](#)  
 nppiCopy\_16s\_C3P3R  
   image\_copy, [766](#)  
 nppiCopy\_16s\_C3R  
   image\_copy, [767](#)  
 nppiCopy\_16s\_C4C1R  
   image\_copy, [767](#)  
 nppiCopy\_16s\_C4CR  
   image\_copy, [767](#)  
 nppiCopy\_16s\_C4MR  
   image\_copy, [768](#)  
 nppiCopy\_16s\_C4P4R  
   image\_copy, [768](#)  
 nppiCopy\_16s\_C4R  
   image\_copy, [768](#)  
 nppiCopy\_16s\_P3C3R  
   image\_copy, [769](#)  
 nppiCopy\_16s\_P4C4R  
   image\_copy, [769](#)  
 nppiCopy\_16sc\_AC4R



image\_copy, [769](#)  
nppiCopy\_16sc\_C1R  
image\_copy, [770](#)  
nppiCopy\_16sc\_C2R  
image\_copy, [770](#)  
nppiCopy\_16sc\_C3R  
image\_copy, [770](#)  
nppiCopy\_16sc\_C4R  
image\_copy, [771](#)  
nppiCopy\_16u\_AC4MR  
image\_copy, [771](#)  
nppiCopy\_16u\_AC4R  
image\_copy, [771](#)  
nppiCopy\_16u\_C1C3R  
image\_copy, [772](#)  
nppiCopy\_16u\_C1C4R  
image\_copy, [772](#)  
nppiCopy\_16u\_C1MR  
image\_copy, [772](#)  
nppiCopy\_16u\_C1R  
image\_copy, [773](#)  
nppiCopy\_16u\_C3C1R  
image\_copy, [773](#)  
nppiCopy\_16u\_C3CR  
image\_copy, [773](#)  
nppiCopy\_16u\_C3MR  
image\_copy, [774](#)  
nppiCopy\_16u\_C3P3R  
image\_copy, [774](#)  
nppiCopy\_16u\_C3R  
image\_copy, [774](#)  
nppiCopy\_16u\_C4C1R  
image\_copy, [775](#)  
nppiCopy\_16u\_C4CR  
image\_copy, [775](#)  
nppiCopy\_16u\_C4MR  
image\_copy, [775](#)  
nppiCopy\_16u\_C4P4R  
image\_copy, [776](#)  
nppiCopy\_16u\_C4R  
image\_copy, [776](#)  
nppiCopy\_16u\_P3C3R  
image\_copy, [776](#)  
nppiCopy\_16u\_P4C4R  
image\_copy, [777](#)  
nppiCopy\_32f\_AC4MR  
image\_copy, [777](#)  
nppiCopy\_32f\_AC4R  
image\_copy, [777](#)  
nppiCopy\_32f\_C1C3R  
image\_copy, [778](#)  
nppiCopy\_32f\_C1C4R  
image\_copy, [778](#)  
nppiCopy\_32f\_C1MR  
image\_copy, [778](#)  
nppiCopy\_32f\_C1R  
image\_copy, [779](#)  
nppiCopy\_32f\_C3C1R  
image\_copy, [779](#)  
nppiCopy\_32f\_C3CR  
image\_copy, [779](#)  
nppiCopy\_32f\_C3MR  
image\_copy, [780](#)  
nppiCopy\_32f\_C3P3R  
image\_copy, [780](#)  
nppiCopy\_32f\_C3R  
image\_copy, [780](#)  
nppiCopy\_32f\_C4C1R  
image\_copy, [781](#)  
nppiCopy\_32f\_C4CR  
image\_copy, [781](#)  
nppiCopy\_32f\_C4MR  
image\_copy, [781](#)  
nppiCopy\_32f\_C4P4R  
image\_copy, [782](#)  
nppiCopy\_32f\_C4R  
image\_copy, [782](#)  
nppiCopy\_32f\_P3C3R  
image\_copy, [782](#)  
nppiCopy\_32f\_P4C4R  
image\_copy, [783](#)  
nppiCopy\_32fc\_AC4R  
image\_copy, [783](#)  
nppiCopy\_32fc\_C1R  
image\_copy, [783](#)  
nppiCopy\_32fc\_C2R  
image\_copy, [784](#)  
nppiCopy\_32fc\_C3R  
image\_copy, [784](#)  
nppiCopy\_32fc\_C4R  
image\_copy, [784](#)  
nppiCopy\_32s\_AC4MR  
image\_copy, [785](#)  
nppiCopy\_32s\_AC4R  
image\_copy, [785](#)  
nppiCopy\_32s\_C1C3R  
image\_copy, [785](#)  
nppiCopy\_32s\_C1C4R  
image\_copy, [786](#)  
nppiCopy\_32s\_C1MR  
image\_copy, [786](#)  
nppiCopy\_32s\_C1R  
image\_copy, [786](#)  
nppiCopy\_32s\_C3C1R  
image\_copy, [787](#)  
nppiCopy\_32s\_C3CR  
image\_copy, [787](#)  
nppiCopy\_32s\_C3MR

image\_copy, 787  
 nppiCopy\_32s\_C3P3R  
   image\_copy, 788  
 nppiCopy\_32s\_C3R  
   image\_copy, 788  
 nppiCopy\_32s\_C4C1R  
   image\_copy, 788  
 nppiCopy\_32s\_C4CR  
   image\_copy, 789  
 nppiCopy\_32s\_C4MR  
   image\_copy, 789  
 nppiCopy\_32s\_C4P4R  
   image\_copy, 789  
 nppiCopy\_32s\_C4R  
   image\_copy, 790  
 nppiCopy\_32s\_P3C3R  
   image\_copy, 790  
 nppiCopy\_32s\_P4C4R  
   image\_copy, 790  
 nppiCopy\_32sc\_AC4R  
   image\_copy, 791  
 nppiCopy\_32sc\_C1R  
   image\_copy, 791  
 nppiCopy\_32sc\_C2R  
   image\_copy, 791  
 nppiCopy\_32sc\_C3R  
   image\_copy, 792  
 nppiCopy\_32sc\_C4R  
   image\_copy, 792  
 nppiCopy\_8s\_AC4R  
   image\_copy, 792  
 nppiCopy\_8s\_C1R  
   image\_copy, 793  
 nppiCopy\_8s\_C2R  
   image\_copy, 793  
 nppiCopy\_8s\_C3R  
   image\_copy, 793  
 nppiCopy\_8s\_C4R  
   image\_copy, 794  
 nppiCopy\_8u\_AC4MR  
   image\_copy, 794  
 nppiCopy\_8u\_AC4R  
   image\_copy, 794  
 nppiCopy\_8u\_C1C3R  
   image\_copy, 795  
 nppiCopy\_8u\_C1C4R  
   image\_copy, 795  
 nppiCopy\_8u\_C1MR  
   image\_copy, 795  
 nppiCopy\_8u\_C1R  
   image\_copy, 796  
 nppiCopy\_8u\_C3C1R  
   image\_copy, 796  
 nppiCopy\_8u\_C3CR

image\_copy, 796  
 nppiCopy\_8u\_C3MR  
   image\_copy, 797  
 nppiCopy\_8u\_C3P3R  
   image\_copy, 797  
 nppiCopy\_8u\_C3R  
   image\_copy, 797  
 nppiCopy\_8u\_C4C1R  
   image\_copy, 798  
 nppiCopy\_8u\_C4CR  
   image\_copy, 798  
 nppiCopy\_8u\_C4MR  
   image\_copy, 798  
 nppiCopy\_8u\_C4P4R  
   image\_copy, 799  
 nppiCopy\_8u\_C4R  
   image\_copy, 799  
 nppiCopy\_8u\_P3C3R  
   image\_copy, 799  
 nppiCopy\_8u\_P4C4R  
   image\_copy, 800  
 nppiCopyConstBorder\_16s\_AC4R  
   image\_copy\_constant\_border, 862  
 nppiCopyConstBorder\_16s\_C1R  
   image\_copy\_constant\_border, 862  
 nppiCopyConstBorder\_16s\_C3R  
   image\_copy\_constant\_border, 863  
 nppiCopyConstBorder\_16s\_C4R  
   image\_copy\_constant\_border, 863  
 nppiCopyConstBorder\_16u\_AC4R  
   image\_copy\_constant\_border, 864  
 nppiCopyConstBorder\_16u\_C1R  
   image\_copy\_constant\_border, 864  
 nppiCopyConstBorder\_16u\_C3R  
   image\_copy\_constant\_border, 865  
 nppiCopyConstBorder\_16u\_C4R  
   image\_copy\_constant\_border, 865  
 nppiCopyConstBorder\_32f\_AC4R  
   image\_copy\_constant\_border, 866  
 nppiCopyConstBorder\_32f\_C1R  
   image\_copy\_constant\_border, 866  
 nppiCopyConstBorder\_32f\_C3R  
   image\_copy\_constant\_border, 867  
 nppiCopyConstBorder\_32f\_C4R  
   image\_copy\_constant\_border, 867  
 nppiCopyConstBorder\_32s\_AC4R  
   image\_copy\_constant\_border, 868  
 nppiCopyConstBorder\_32s\_C1R  
   image\_copy\_constant\_border, 868  
 nppiCopyConstBorder\_32s\_C3R  
   image\_copy\_constant\_border, 869  
 nppiCopyConstBorder\_32s\_C4R  
   image\_copy\_constant\_border, 869  
 nppiCopyConstBorder\_8u\_AC4R

- image\_copy\_constant\_border, 870
- nppiCopyConstBorder\_8u\_C1R
  - image\_copy\_constant\_border, 870
- nppiCopyConstBorder\_8u\_C3R
  - image\_copy\_constant\_border, 871
- nppiCopyConstBorder\_8u\_C4R
  - image\_copy\_constant\_border, 871
- nppiCopyReplicateBorder\_16s\_AC4R
  - image\_copy\_replicate\_border, 875
- nppiCopyReplicateBorder\_16s\_C1R
  - image\_copy\_replicate\_border, 875
- nppiCopyReplicateBorder\_16s\_C3R
  - image\_copy\_replicate\_border, 876
- nppiCopyReplicateBorder\_16s\_C4R
  - image\_copy\_replicate\_border, 876
- nppiCopyReplicateBorder\_16u\_AC4R
  - image\_copy\_replicate\_border, 877
- nppiCopyReplicateBorder\_16u\_C1R
  - image\_copy\_replicate\_border, 877
- nppiCopyReplicateBorder\_16u\_C3R
  - image\_copy\_replicate\_border, 878
- nppiCopyReplicateBorder\_16u\_C4R
  - image\_copy\_replicate\_border, 878
- nppiCopyReplicateBorder\_32f\_AC4R
  - image\_copy\_replicate\_border, 878
- nppiCopyReplicateBorder\_32f\_C1R
  - image\_copy\_replicate\_border, 879
- nppiCopyReplicateBorder\_32f\_C3R
  - image\_copy\_replicate\_border, 879
- nppiCopyReplicateBorder\_32f\_C4R
  - image\_copy\_replicate\_border, 880
- nppiCopyReplicateBorder\_32s\_AC4R
  - image\_copy\_replicate\_border, 880
- nppiCopyReplicateBorder\_32s\_C1R
  - image\_copy\_replicate\_border, 881
- nppiCopyReplicateBorder\_32s\_C3R
  - image\_copy\_replicate\_border, 881
- nppiCopyReplicateBorder\_32s\_C4R
  - image\_copy\_replicate\_border, 882
- nppiCopyReplicateBorder\_8u\_AC4R
  - image\_copy\_replicate\_border, 882
- nppiCopyReplicateBorder\_8u\_C1R
  - image\_copy\_replicate\_border, 883
- nppiCopyReplicateBorder\_8u\_C3R
  - image\_copy\_replicate\_border, 883
- nppiCopyReplicateBorder\_8u\_C4R
  - image\_copy\_replicate\_border, 884
- nppiCopySubpix\_16s\_AC4R
  - image\_copy\_sub\_pixel, 899
- nppiCopySubpix\_16s\_C1R
  - image\_copy\_sub\_pixel, 900
- nppiCopySubpix\_16s\_C3R
  - image\_copy\_sub\_pixel, 900
- nppiCopySubpix\_16s\_C4R
  - image\_copy\_sub\_pixel, 901
- nppiCopySubpix\_16u\_AC4R
  - image\_copy\_sub\_pixel, 901
- nppiCopySubpix\_16u\_C1R
  - image\_copy\_sub\_pixel, 901
- nppiCopySubpix\_16u\_C3R
  - image\_copy\_sub\_pixel, 902
- nppiCopySubpix\_16u\_C4R
  - image\_copy\_sub\_pixel, 902
- nppiCopySubpix\_32f\_AC4R
  - image\_copy\_sub\_pixel, 903
- nppiCopySubpix\_32f\_C1R
  - image\_copy\_sub\_pixel, 903
- nppiCopySubpix\_32f\_C3R
  - image\_copy\_sub\_pixel, 903
- nppiCopySubpix\_32f\_C4R
  - image\_copy\_sub\_pixel, 904
- nppiCopySubpix\_32s\_AC4R
  - image\_copy\_sub\_pixel, 904
- nppiCopySubpix\_32s\_C1R
  - image\_copy\_sub\_pixel, 905
- nppiCopySubpix\_32s\_C3R
  - image\_copy\_sub\_pixel, 905
- nppiCopySubpix\_32s\_C4R
  - image\_copy\_sub\_pixel, 906
- nppiCopySubpix\_8u\_AC4R
  - image\_copy\_sub\_pixel, 906
- nppiCopySubpix\_8u\_C1R
  - image\_copy\_sub\_pixel, 906
- nppiCopySubpix\_8u\_C3R
  - image\_copy\_sub\_pixel, 907
- nppiCopySubpix\_8u\_C4R
  - image\_copy\_sub\_pixel, 907
- nppiCopyWrapBorder\_16s\_AC4R
  - image\_copy\_wrap\_border, 887
- nppiCopyWrapBorder\_16s\_C1R
  - image\_copy\_wrap\_border, 887
- nppiCopyWrapBorder\_16s\_C3R
  - image\_copy\_wrap\_border, 888
- nppiCopyWrapBorder\_16s\_C4R
  - image\_copy\_wrap\_border, 888
- nppiCopyWrapBorder\_16u\_AC4R
  - image\_copy\_wrap\_border, 889
- nppiCopyWrapBorder\_16u\_C1R
  - image\_copy\_wrap\_border, 889
- nppiCopyWrapBorder\_16u\_C3R
  - image\_copy\_wrap\_border, 890
- nppiCopyWrapBorder\_16u\_C4R
  - image\_copy\_wrap\_border, 890
- nppiCopyWrapBorder\_32f\_AC4R
  - image\_copy\_wrap\_border, 891
- nppiCopyWrapBorder\_32f\_C1R
  - image\_copy\_wrap\_border, 891
- nppiCopyWrapBorder\_32f\_C3R

- image\_copy\_wrap\_border, [892](#)
- nppiCopyWrapBorder\_32f\_C4R
  - image\_copy\_wrap\_border, [892](#)
- nppiCopyWrapBorder\_32s\_AC4R
  - image\_copy\_wrap\_border, [893](#)
- nppiCopyWrapBorder\_32s\_C1R
  - image\_copy\_wrap\_border, [893](#)
- nppiCopyWrapBorder\_32s\_C3R
  - image\_copy\_wrap\_border, [894](#)
- nppiCopyWrapBorder\_32s\_C4R
  - image\_copy\_wrap\_border, [894](#)
- nppiCopyWrapBorder\_8u\_AC4R
  - image\_copy\_wrap\_border, [895](#)
- nppiCopyWrapBorder\_8u\_C1R
  - image\_copy\_wrap\_border, [895](#)
- nppiCopyWrapBorder\_8u\_C3R
  - image\_copy\_wrap\_border, [896](#)
- nppiCopyWrapBorder\_8u\_C4R
  - image\_copy\_wrap\_border, [896](#)
- nppiCountInRange\_32f\_AC4R
  - image\_count\_in\_range, [1827](#)
- nppiCountInRange\_32f\_C1R
  - image\_count\_in\_range, [1827](#)
- nppiCountInRange\_32f\_C3R
  - image\_count\_in\_range, [1828](#)
- nppiCountInRange\_8u\_AC4R
  - image\_count\_in\_range, [1828](#)
- nppiCountInRange\_8u\_C1R
  - image\_count\_in\_range, [1829](#)
- nppiCountInRange\_8u\_C3R
  - image\_count\_in\_range, [1829](#)
- nppiCountInRangeGetBufferHostSize\_32f\_AC4R
  - image\_count\_in\_range, [1830](#)
- nppiCountInRangeGetBufferHostSize\_32f\_C1R
  - image\_count\_in\_range, [1830](#)
- nppiCountInRangeGetBufferHostSize\_32f\_C3R
  - image\_count\_in\_range, [1830](#)
- nppiCountInRangeGetBufferHostSize\_8u\_AC4R
  - image\_count\_in\_range, [1830](#)
- nppiCountInRangeGetBufferHostSize\_8u\_C1R
  - image\_count\_in\_range, [1831](#)
- nppiCountInRangeGetBufferHostSize\_8u\_C3R
  - image\_count\_in\_range, [1831](#)
- nppiCrossCorrFull\_Norm\_16u32f\_AC4R
  - crosscorrfullnorm, [1921](#)
- nppiCrossCorrFull\_Norm\_16u32f\_C1R
  - crosscorrfullnorm, [1921](#)
- nppiCrossCorrFull\_Norm\_16u32f\_C3R
  - crosscorrfullnorm, [1921](#)
- nppiCrossCorrFull\_Norm\_16u32f\_C4R
  - crosscorrfullnorm, [1922](#)
- nppiCrossCorrFull\_Norm\_32f\_AC4R
  - crosscorrfullnorm, [1922](#)
- nppiCrossCorrFull\_Norm\_32f\_C1R
  - crosscorrfullnorm, [1923](#)
- nppiCrossCorrFull\_Norm\_32f\_C3R
  - crosscorrfullnorm, [1923](#)
- nppiCrossCorrFull\_Norm\_32f\_C4R
  - crosscorrfullnorm, [1924](#)
- nppiCrossCorrFull\_Norm\_8s32f\_AC4R
  - crosscorrfullnorm, [1924](#)
- nppiCrossCorrFull\_Norm\_8s32f\_C1R
  - crosscorrfullnorm, [1924](#)
- nppiCrossCorrFull\_Norm\_8s32f\_C3R
  - crosscorrfullnorm, [1925](#)
- nppiCrossCorrFull\_Norm\_8s32f\_C4R
  - crosscorrfullnorm, [1925](#)
- nppiCrossCorrFull\_Norm\_8u32f\_AC4R
  - crosscorrfullnorm, [1926](#)
- nppiCrossCorrFull\_Norm\_8u32f\_C1R
  - crosscorrfullnorm, [1926](#)
- nppiCrossCorrFull\_Norm\_8u32f\_C3R
  - crosscorrfullnorm, [1927](#)
- nppiCrossCorrFull\_Norm\_8u32f\_C4R
  - crosscorrfullnorm, [1927](#)
- nppiCrossCorrFull\_Norm\_8u\_AC4RSfs
  - crosscorrfullnorm, [1927](#)
- nppiCrossCorrFull\_Norm\_8u\_C1RSfs
  - crosscorrfullnorm, [1928](#)
- nppiCrossCorrFull\_Norm\_8u\_C3RSfs
  - crosscorrfullnorm, [1928](#)
- nppiCrossCorrFull\_Norm\_8u\_C4RSfs
  - crosscorrfullnorm, [1929](#)
- nppiCrossCorrFull\_NormLevel\_16u32f\_AC4R
  - crosscorrfullnormlevel, [1959](#)
- nppiCrossCorrFull\_NormLevel\_16u32f\_C1R
  - crosscorrfullnormlevel, [1959](#)
- nppiCrossCorrFull\_NormLevel\_16u32f\_C3R
  - crosscorrfullnormlevel, [1959](#)
- nppiCrossCorrFull\_NormLevel\_16u32f\_C4R
  - crosscorrfullnormlevel, [1960](#)
- nppiCrossCorrFull\_NormLevel\_32f\_AC4R
  - crosscorrfullnormlevel, [1960](#)
- nppiCrossCorrFull\_NormLevel\_32f\_C1R
  - crosscorrfullnormlevel, [1961](#)
- nppiCrossCorrFull\_NormLevel\_32f\_C3R
  - crosscorrfullnormlevel, [1961](#)
- nppiCrossCorrFull\_NormLevel\_32f\_C4R
  - crosscorrfullnormlevel, [1962](#)
- nppiCrossCorrFull\_NormLevel\_8s32f\_AC4R
  - crosscorrfullnormlevel, [1962](#)
- nppiCrossCorrFull\_NormLevel\_8s32f\_C1R
  - crosscorrfullnormlevel, [1963](#)
- nppiCrossCorrFull\_NormLevel\_8s32f\_C3R
  - crosscorrfullnormlevel, [1963](#)
- nppiCrossCorrFull\_NormLevel\_8s32f\_C4R
  - crosscorrfullnormlevel, [1964](#)
- nppiCrossCorrFull\_NormLevel\_8u32f\_AC4R
  - crosscorrfullnormlevel, [1964](#)

- crosscorrfullnormlevel, [1964](#)
- nppiCrossCorrFull\_NormLevel\_8u32f\_C1R
  - crosscorrfullnormlevel, [1965](#)
- nppiCrossCorrFull\_NormLevel\_8u32f\_C3R
  - crosscorrfullnormlevel, [1965](#)
- nppiCrossCorrFull\_NormLevel\_8u32f\_C4R
  - crosscorrfullnormlevel, [1966](#)
- nppiCrossCorrFull\_NormLevel\_8u\_AC4RSfs
  - crosscorrfullnormlevel, [1966](#)
- nppiCrossCorrFull\_NormLevel\_8u\_C1RSfs
  - crosscorrfullnormlevel, [1967](#)
- nppiCrossCorrFull\_NormLevel\_8u\_C3RSfs
  - crosscorrfullnormlevel, [1967](#)
- nppiCrossCorrFull\_NormLevel\_8u\_C4RSfs
  - crosscorrfullnormlevel, [1968](#)
- nppiCrossCorrSame\_Norm\_16u32f\_AC4R
  - crosscorrssamenorm, [1932](#)
- nppiCrossCorrSame\_Norm\_16u32f\_C1R
  - crosscorrssamenorm, [1932](#)
- nppiCrossCorrSame\_Norm\_16u32f\_C3R
  - crosscorrssamenorm, [1932](#)
- nppiCrossCorrSame\_Norm\_16u32f\_C4R
  - crosscorrssamenorm, [1933](#)
- nppiCrossCorrSame\_Norm\_32f\_AC4R
  - crosscorrssamenorm, [1933](#)
- nppiCrossCorrSame\_Norm\_32f\_C1R
  - crosscorrssamenorm, [1934](#)
- nppiCrossCorrSame\_Norm\_32f\_C3R
  - crosscorrssamenorm, [1934](#)
- nppiCrossCorrSame\_Norm\_32f\_C4R
  - crosscorrssamenorm, [1935](#)
- nppiCrossCorrSame\_Norm\_8s32f\_AC4R
  - crosscorrssamenorm, [1935](#)
- nppiCrossCorrSame\_Norm\_8s32f\_C1R
  - crosscorrssamenorm, [1935](#)
- nppiCrossCorrSame\_Norm\_8s32f\_C3R
  - crosscorrssamenorm, [1936](#)
- nppiCrossCorrSame\_Norm\_8s32f\_C4R
  - crosscorrssamenorm, [1936](#)
- nppiCrossCorrSame\_Norm\_8u32f\_AC4R
  - crosscorrssamenorm, [1937](#)
- nppiCrossCorrSame\_Norm\_8u32f\_C1R
  - crosscorrssamenorm, [1937](#)
- nppiCrossCorrSame\_Norm\_8u32f\_C3R
  - crosscorrssamenorm, [1938](#)
- nppiCrossCorrSame\_Norm\_8u32f\_C4R
  - crosscorrssamenorm, [1938](#)
- nppiCrossCorrSame\_Norm\_8u\_AC4RSfs
  - crosscorrssamenorm, [1938](#)
- nppiCrossCorrSame\_Norm\_8u\_C1RSfs
  - crosscorrssamenorm, [1939](#)
- nppiCrossCorrSame\_Norm\_8u\_C3RSfs
  - crosscorrssamenorm, [1939](#)
- nppiCrossCorrSame\_Norm\_8u\_C4RSfs
  - crosscorrssamenorm, [1940](#)
- nppiCrossCorrSame\_NormLevel\_16u32f\_AC4R
  - crosscorrssamenormlevel, [1979](#)
- nppiCrossCorrSame\_NormLevel\_16u32f\_C1R
  - crosscorrssamenormlevel, [1979](#)
- nppiCrossCorrSame\_NormLevel\_16u32f\_C3R
  - crosscorrssamenormlevel, [1979](#)
- nppiCrossCorrSame\_NormLevel\_16u32f\_C4R
  - crosscorrssamenormlevel, [1980](#)
- nppiCrossCorrSame\_NormLevel\_32f\_AC4R
  - crosscorrssamenormlevel, [1980](#)
- nppiCrossCorrSame\_NormLevel\_32f\_C1R
  - crosscorrssamenormlevel, [1981](#)
- nppiCrossCorrSame\_NormLevel\_32f\_C3R
  - crosscorrssamenormlevel, [1981](#)
- nppiCrossCorrSame\_NormLevel\_32f\_C4R
  - crosscorrssamenormlevel, [1982](#)
- nppiCrossCorrSame\_NormLevel\_8s32f\_AC4R
  - crosscorrssamenormlevel, [1982](#)
- nppiCrossCorrSame\_NormLevel\_8s32f\_C1R
  - crosscorrssamenormlevel, [1983](#)
- nppiCrossCorrSame\_NormLevel\_8s32f\_C3R
  - crosscorrssamenormlevel, [1983](#)
- nppiCrossCorrSame\_NormLevel\_8s32f\_C4R
  - crosscorrssamenormlevel, [1984](#)
- nppiCrossCorrSame\_NormLevel\_8u32f\_AC4R
  - crosscorrssamenormlevel, [1984](#)
- nppiCrossCorrSame\_NormLevel\_8u32f\_C1R
  - crosscorrssamenormlevel, [1985](#)
- nppiCrossCorrSame\_NormLevel\_8u32f\_C3R
  - crosscorrssamenormlevel, [1985](#)
- nppiCrossCorrSame\_NormLevel\_8u32f\_C4R
  - crosscorrssamenormlevel, [1986](#)
- nppiCrossCorrSame\_NormLevel\_8u\_AC4RSfs
  - crosscorrssamenormlevel, [1986](#)
- nppiCrossCorrSame\_NormLevel\_8u\_C1RSfs
  - crosscorrssamenormlevel, [1987](#)
- nppiCrossCorrSame\_NormLevel\_8u\_C3RSfs
  - crosscorrssamenormlevel, [1987](#)
- nppiCrossCorrSame\_NormLevel\_8u\_C4RSfs
  - crosscorrssamenormlevel, [1988](#)
- nppiCrossCorrValid\_16u32f\_C1R
  - crosscorrvalid, [1952](#)
- nppiCrossCorrValid\_32f\_C1R
  - crosscorrvalid, [1953](#)
- nppiCrossCorrValid\_8s32f\_C1R
  - crosscorrvalid, [1953](#)
- nppiCrossCorrValid\_8u32f\_C1R
  - crosscorrvalid, [1953](#)
- nppiCrossCorrValid\_Norm\_16u32f\_AC4R
  - crosscorrvalidnorm, [1943](#)
- nppiCrossCorrValid\_Norm\_16u32f\_C1R
  - crosscorrvalidnorm, [1943](#)
- nppiCrossCorrValid\_Norm\_16u32f\_C3R

- crosscorrvalidnorm, [1943](#)
- nppiCrossCorrValid\_Norm\_16u32f\_C4R
  - crosscorrvalidnorm, [1944](#)
- nppiCrossCorrValid\_Norm\_32f\_AC4R
  - crosscorrvalidnorm, [1944](#)
- nppiCrossCorrValid\_Norm\_32f\_C1R
  - crosscorrvalidnorm, [1945](#)
- nppiCrossCorrValid\_Norm\_32f\_C3R
  - crosscorrvalidnorm, [1945](#)
- nppiCrossCorrValid\_Norm\_32f\_C4R
  - crosscorrvalidnorm, [1946](#)
- nppiCrossCorrValid\_Norm\_8s32f\_AC4R
  - crosscorrvalidnorm, [1946](#)
- nppiCrossCorrValid\_Norm\_8s32f\_C1R
  - crosscorrvalidnorm, [1946](#)
- nppiCrossCorrValid\_Norm\_8s32f\_C3R
  - crosscorrvalidnorm, [1947](#)
- nppiCrossCorrValid\_Norm\_8s32f\_C4R
  - crosscorrvalidnorm, [1947](#)
- nppiCrossCorrValid\_Norm\_8u32f\_AC4R
  - crosscorrvalidnorm, [1948](#)
- nppiCrossCorrValid\_Norm\_8u32f\_C1R
  - crosscorrvalidnorm, [1948](#)
- nppiCrossCorrValid\_Norm\_8u32f\_C3R
  - crosscorrvalidnorm, [1949](#)
- nppiCrossCorrValid\_Norm\_8u32f\_C4R
  - crosscorrvalidnorm, [1949](#)
- nppiCrossCorrValid\_Norm\_8u\_AC4RSfs
  - crosscorrvalidnorm, [1949](#)
- nppiCrossCorrValid\_Norm\_8u\_C1RSfs
  - crosscorrvalidnorm, [1950](#)
- nppiCrossCorrValid\_Norm\_8u\_C3RSfs
  - crosscorrvalidnorm, [1950](#)
- nppiCrossCorrValid\_Norm\_8u\_C4RSfs
  - crosscorrvalidnorm, [1951](#)
- nppiCrossCorrValid\_NormLevel\_16u32f\_AC4R
  - crosscorrvalidnormlevel, [1999](#)
- nppiCrossCorrValid\_NormLevel\_16u32f\_C1R
  - crosscorrvalidnormlevel, [1999](#)
- nppiCrossCorrValid\_NormLevel\_16u32f\_C3R
  - crosscorrvalidnormlevel, [1999](#)
- nppiCrossCorrValid\_NormLevel\_16u32f\_C4R
  - crosscorrvalidnormlevel, [2000](#)
- nppiCrossCorrValid\_NormLevel\_32f\_AC4R
  - crosscorrvalidnormlevel, [2000](#)
- nppiCrossCorrValid\_NormLevel\_32f\_C1R
  - crosscorrvalidnormlevel, [2001](#)
- nppiCrossCorrValid\_NormLevel\_32f\_C3R
  - crosscorrvalidnormlevel, [2001](#)
- nppiCrossCorrValid\_NormLevel\_32f\_C4R
  - crosscorrvalidnormlevel, [2002](#)
- nppiCrossCorrValid\_NormLevel\_8s32f\_AC4R
  - crosscorrvalidnormlevel, [2002](#)
- nppiCrossCorrValid\_NormLevel\_8s32f\_C1R
  - crosscorrvalidnormlevel, [2003](#)
- nppiCrossCorrValid\_NormLevel\_8s32f\_C3R
  - crosscorrvalidnormlevel, [2003](#)
- nppiCrossCorrValid\_NormLevel\_8s32f\_C4R
  - crosscorrvalidnormlevel, [2004](#)
- nppiCrossCorrValid\_NormLevel\_8u32f\_AC4R
  - crosscorrvalidnormlevel, [2004](#)
- nppiCrossCorrValid\_NormLevel\_8u32f\_C1R
  - crosscorrvalidnormlevel, [2005](#)
- nppiCrossCorrValid\_NormLevel\_8u32f\_C3R
  - crosscorrvalidnormlevel, [2005](#)
- nppiCrossCorrValid\_NormLevel\_8u32f\_C4R
  - crosscorrvalidnormlevel, [2006](#)
- nppiCrossCorrValid\_NormLevel\_8u\_AC4RSfs
  - crosscorrvalidnormlevel, [2006](#)
- nppiCrossCorrValid\_NormLevel\_8u\_C1RSfs
  - crosscorrvalidnormlevel, [2007](#)
- nppiCrossCorrValid\_NormLevel\_8u\_C3RSfs
  - crosscorrvalidnormlevel, [2007](#)
- nppiCrossCorrValid\_NormLevel\_8u\_C4RSfs
  - crosscorrvalidnormlevel, [2008](#)
- nppiDCTable
  - typedefs\_npp, [42](#)
- nppiDCTFree
  - image\_quantization, [706](#)
- nppiDCTInitAlloc
  - image\_quantization, [706](#)
- nppiDCTQuantFwd8x8LS\_JPEG\_8u16s\_C1R
  - image\_quantization, [706](#)
- nppiDCTQuantFwd8x8LS\_JPEG\_8u16s\_C1R\_-NEW
  - image\_quantization, [707](#)
- nppiDCTQuantInv8x8LS\_JPEG\_16s8u\_C1R
  - image\_quantization, [707](#)
- nppiDCTQuantInv8x8LS\_JPEG\_16s8u\_C1R\_-NEW
  - image\_quantization, [708](#)
- NppiDCTState
  - image\_quantization, [706](#)
- nppiDecodeHuffmanScanHost\_JPEG\_8u16s\_P1R
  - image\_compression, [702](#)
- nppiDecodeHuffmanScanHost\_JPEG\_8u16s\_P3R
  - image\_compression, [702](#)
- NppiDecodeHuffmanSpec
  - image\_compression, [702](#)
- nppiDecodeHuffmanSpecFreeHost\_JPEG
  - image\_compression, [703](#)
- nppiDecodeHuffmanSpecGetBufSize\_JPEG
  - image\_compression, [703](#)
- nppiDecodeHuffmanSpecInitAllocHost\_JPEG
  - image\_compression, [703](#)
- nppiDecodeHuffmanSpecInitHost\_JPEG
  - image\_compression, [704](#)
- nppiDilate3x3\_16u\_AC4R



image\_dilate\_3x3, [1382](#)  
nppiDilate3x3\_16u\_C1R  
image\_dilate\_3x3, [1382](#)  
nppiDilate3x3\_16u\_C3R  
image\_dilate\_3x3, [1382](#)  
nppiDilate3x3\_16u\_C4R  
image\_dilate\_3x3, [1383](#)  
nppiDilate3x3\_32f\_AC4R  
image\_dilate\_3x3, [1383](#)  
nppiDilate3x3\_32f\_C1R  
image\_dilate\_3x3, [1383](#)  
nppiDilate3x3\_32f\_C3R  
image\_dilate\_3x3, [1384](#)  
nppiDilate3x3\_32f\_C4R  
image\_dilate\_3x3, [1384](#)  
nppiDilate3x3\_64f\_C1R  
image\_dilate\_3x3, [1384](#)  
nppiDilate3x3\_8u\_AC4R  
image\_dilate\_3x3, [1385](#)  
nppiDilate3x3\_8u\_C1R  
image\_dilate\_3x3, [1385](#)  
nppiDilate3x3\_8u\_C3R  
image\_dilate\_3x3, [1385](#)  
nppiDilate3x3\_8u\_C4R  
image\_dilate\_3x3, [1386](#)  
nppiDilate\_16u\_AC4R  
image\_dilate, [1368](#)  
nppiDilate\_16u\_C1R  
image\_dilate, [1368](#)  
nppiDilate\_16u\_C3R  
image\_dilate, [1369](#)  
nppiDilate\_16u\_C4R  
image\_dilate, [1369](#)  
nppiDilate\_32f\_AC4R  
image\_dilate, [1369](#)  
nppiDilate\_32f\_C1R  
image\_dilate, [1370](#)  
nppiDilate\_32f\_C3R  
image\_dilate, [1370](#)  
nppiDilate\_32f\_C4R  
image\_dilate, [1371](#)  
nppiDilate\_8u\_AC4R  
image\_dilate, [1371](#)  
nppiDilate\_8u\_C1R  
image\_dilate, [1372](#)  
nppiDilate\_8u\_C3R  
image\_dilate, [1372](#)  
nppiDilate\_8u\_C4R  
image\_dilate, [1372](#)  
nppiDiv\_16s\_AC4IRSfs  
image\_div, [281](#)  
nppiDiv\_16s\_AC4RSfs  
image\_div, [281](#)  
nppiDiv\_16s\_C1IRSfs  
image\_div, [282](#)  
nppiDiv\_16s\_C1RSfs  
image\_div, [282](#)  
nppiDiv\_16s\_C3IRSfs  
image\_div, [282](#)  
nppiDiv\_16s\_C3RSfs  
image\_div, [283](#)  
nppiDiv\_16s\_C4IRSfs  
image\_div, [283](#)  
nppiDiv\_16s\_C4RSfs  
image\_div, [284](#)  
nppiDiv\_16sc\_AC4IRSfs  
image\_div, [284](#)  
nppiDiv\_16sc\_AC4RSfs  
image\_div, [284](#)  
nppiDiv\_16sc\_C1IRSfs  
image\_div, [285](#)  
nppiDiv\_16sc\_C1RSfs  
image\_div, [285](#)  
nppiDiv\_16sc\_C3IRSfs  
image\_div, [286](#)  
nppiDiv\_16sc\_C3RSfs  
image\_div, [286](#)  
nppiDiv\_16u\_AC4IRSfs  
image\_div, [287](#)  
nppiDiv\_16u\_AC4RSfs  
image\_div, [287](#)  
nppiDiv\_16u\_C1IRSfs  
image\_div, [287](#)  
nppiDiv\_16u\_C1RSfs  
image\_div, [288](#)  
nppiDiv\_16u\_C3IRSfs  
image\_div, [288](#)  
nppiDiv\_16u\_C3RSfs  
image\_div, [289](#)  
nppiDiv\_16u\_C4IRSfs  
image\_div, [289](#)  
nppiDiv\_16u\_C4RSfs  
image\_div, [289](#)  
nppiDiv\_32f\_AC4IR  
image\_div, [290](#)  
nppiDiv\_32f\_AC4R  
image\_div, [290](#)  
nppiDiv\_32f\_C1IR  
image\_div, [291](#)  
nppiDiv\_32f\_C1R  
image\_div, [291](#)  
nppiDiv\_32f\_C3IR  
image\_div, [291](#)  
nppiDiv\_32f\_C3R  
image\_div, [292](#)  
nppiDiv\_32f\_C4IR  
image\_div, [292](#)  
nppiDiv\_32f\_C4R

- image\_div, [292](#)
- nppiDiv\_32fc\_AC4IR
  - image\_div, [293](#)
- nppiDiv\_32fc\_AC4R
  - image\_div, [293](#)
- nppiDiv\_32fc\_C1IR
  - image\_div, [294](#)
- nppiDiv\_32fc\_C1R
  - image\_div, [294](#)
- nppiDiv\_32fc\_C3IR
  - image\_div, [294](#)
- nppiDiv\_32fc\_C3R
  - image\_div, [295](#)
- nppiDiv\_32fc\_C4IR
  - image\_div, [295](#)
- nppiDiv\_32fc\_C4R
  - image\_div, [295](#)
- nppiDiv\_32s\_C1IRSfs
  - image\_div, [296](#)
- nppiDiv\_32s\_C1R
  - image\_div, [296](#)
- nppiDiv\_32s\_C1RSfs
  - image\_div, [297](#)
- nppiDiv\_32s\_C3IRSfs
  - image\_div, [297](#)
- nppiDiv\_32s\_C3RSfs
  - image\_div, [297](#)
- nppiDiv\_32sc\_AC4IRSfs
  - image\_div, [298](#)
- nppiDiv\_32sc\_AC4RSfs
  - image\_div, [298](#)
- nppiDiv\_32sc\_C1IRSfs
  - image\_div, [299](#)
- nppiDiv\_32sc\_C1RSfs
  - image\_div, [299](#)
- nppiDiv\_32sc\_C3IRSfs
  - image\_div, [300](#)
- nppiDiv\_32sc\_C3RSfs
  - image\_div, [300](#)
- nppiDiv\_8u\_AC4IRSfs
  - image\_div, [300](#)
- nppiDiv\_8u\_AC4RSfs
  - image\_div, [301](#)
- nppiDiv\_8u\_C1IRSfs
  - image\_div, [301](#)
- nppiDiv\_8u\_C1RSfs
  - image\_div, [302](#)
- nppiDiv\_8u\_C3IRSfs
  - image\_div, [302](#)
- nppiDiv\_8u\_C3RSfs
  - image\_div, [302](#)
- nppiDiv\_8u\_C4IRSfs
  - image\_div, [303](#)
- nppiDiv\_8u\_C4RSfs
  - image\_div, [303](#)
- image\_div, [303](#)
- nppiDiv\_Round\_16s\_AC4IRSfs
  - image\_divround, [307](#)
- nppiDiv\_Round\_16s\_AC4RSfs
  - image\_divround, [308](#)
- nppiDiv\_Round\_16s\_C1IRSfs
  - image\_divround, [308](#)
- nppiDiv\_Round\_16s\_C1RSfs
  - image\_divround, [309](#)
- nppiDiv\_Round\_16s\_C3IRSfs
  - image\_divround, [309](#)
- nppiDiv\_Round\_16s\_C3RSfs
  - image\_divround, [309](#)
- nppiDiv\_Round\_16s\_C4IRSfs
  - image\_divround, [310](#)
- nppiDiv\_Round\_16s\_C4RSfs
  - image\_divround, [310](#)
- nppiDiv\_Round\_16u\_AC4IRSfs
  - image\_divround, [311](#)
- nppiDiv\_Round\_16u\_AC4RSfs
  - image\_divround, [311](#)
- nppiDiv\_Round\_16u\_C1IRSfs
  - image\_divround, [312](#)
- nppiDiv\_Round\_16u\_C1RSfs
  - image\_divround, [312](#)
- nppiDiv\_Round\_16u\_C3IRSfs
  - image\_divround, [313](#)
- nppiDiv\_Round\_16u\_C3RSfs
  - image\_divround, [313](#)
- nppiDiv\_Round\_16u\_C4IRSfs
  - image\_divround, [314](#)
- nppiDiv\_Round\_16u\_C4RSfs
  - image\_divround, [314](#)
- nppiDiv\_Round\_8u\_AC4IRSfs
  - image\_divround, [315](#)
- nppiDiv\_Round\_8u\_AC4RSfs
  - image\_divround, [315](#)
- nppiDiv\_Round\_8u\_C1IRSfs
  - image\_divround, [316](#)
- nppiDiv\_Round\_8u\_C1RSfs
  - image\_divround, [316](#)
- nppiDiv\_Round\_8u\_C3IRSfs
  - image\_divround, [317](#)
- nppiDiv\_Round\_8u\_C3RSfs
  - image\_divround, [317](#)
- nppiDiv\_Round\_8u\_C4IRSfs
  - image\_divround, [318](#)
- nppiDiv\_Round\_8u\_C4RSfs
  - image\_divround, [318](#)
- nppiDivC\_16s\_AC4IRSfs
  - image\_divc, [145](#)
- nppiDivC\_16s\_AC4RSfs
  - image\_divc, [145](#)
- nppiDivC\_16s\_C1IRSfs



- image\_divc, [145](#)
- nppiDivC\_16s\_C1RSfs
  - image\_divc, [146](#)
- nppiDivC\_16s\_C3RSfs
  - image\_divc, [146](#)
- nppiDivC\_16s\_C3RSfs
  - image\_divc, [146](#)
- nppiDivC\_16s\_C4RSfs
  - image\_divc, [147](#)
- nppiDivC\_16s\_C4RSfs
  - image\_divc, [147](#)
- nppiDivC\_16sc\_AC4RSfs
  - image\_divc, [148](#)
- nppiDivC\_16sc\_AC4RSfs
  - image\_divc, [148](#)
- nppiDivC\_16sc\_C1RSfs
  - image\_divc, [148](#)
- nppiDivC\_16sc\_C1RSfs
  - image\_divc, [149](#)
- nppiDivC\_16sc\_C3RSfs
  - image\_divc, [149](#)
- nppiDivC\_16sc\_C3RSfs
  - image\_divc, [150](#)
- nppiDivC\_16u\_AC4RSfs
  - image\_divc, [150](#)
- nppiDivC\_16u\_AC4RSfs
  - image\_divc, [150](#)
- nppiDivC\_16u\_C1RSfs
  - image\_divc, [151](#)
- nppiDivC\_16u\_C1RSfs
  - image\_divc, [151](#)
- nppiDivC\_16u\_C3RSfs
  - image\_divc, [152](#)
- nppiDivC\_16u\_C3RSfs
  - image\_divc, [152](#)
- nppiDivC\_16u\_C4RSfs
  - image\_divc, [153](#)
- nppiDivC\_32f\_AC4IR
  - image\_divc, [153](#)
- nppiDivC\_32f\_AC4R
  - image\_divc, [153](#)
- nppiDivC\_32f\_C1IR
  - image\_divc, [154](#)
- nppiDivC\_32f\_C1R
  - image\_divc, [154](#)
- nppiDivC\_32f\_C3IR
  - image\_divc, [154](#)
- nppiDivC\_32f\_C3R
  - image\_divc, [155](#)
- nppiDivC\_32f\_C4IR
  - image\_divc, [155](#)
- nppiDivC\_32f\_C4R
  - image\_divc, [155](#)
- nppiDivC\_32fc\_AC4IR
  - image\_divc, [156](#)
- nppiDivC\_32fc\_AC4R
  - image\_divc, [156](#)
- nppiDivC\_32fc\_C1IR
  - image\_divc, [156](#)
- nppiDivC\_32fc\_C1R
  - image\_divc, [157](#)
- nppiDivC\_32fc\_C3IR
  - image\_divc, [157](#)
- nppiDivC\_32fc\_C3R
  - image\_divc, [157](#)
- nppiDivC\_32fc\_C4IR
  - image\_divc, [158](#)
- nppiDivC\_32fc\_C4R
  - image\_divc, [158](#)
- nppiDivC\_32s\_C1RSfs
  - image\_divc, [159](#)
- nppiDivC\_32s\_C1RSfs
  - image\_divc, [159](#)
- nppiDivC\_32s\_C3RSfs
  - image\_divc, [159](#)
- nppiDivC\_32s\_C3RSfs
  - image\_divc, [160](#)
- nppiDivC\_32sc\_AC4RSfs
  - image\_divc, [160](#)
- nppiDivC\_32sc\_AC4RSfs
  - image\_divc, [160](#)
- nppiDivC\_32sc\_C1RSfs
  - image\_divc, [161](#)
- nppiDivC\_32sc\_C1RSfs
  - image\_divc, [161](#)
- nppiDivC\_32sc\_C3RSfs
  - image\_divc, [162](#)
- nppiDivC\_32sc\_C3RSfs
  - image\_divc, [162](#)
- nppiDivC\_8u\_AC4RSfs
  - image\_divc, [162](#)
- nppiDivC\_8u\_AC4RSfs
  - image\_divc, [163](#)
- nppiDivC\_8u\_C1RSfs
  - image\_divc, [163](#)
- nppiDivC\_8u\_C1RSfs
  - image\_divc, [164](#)
- nppiDivC\_8u\_C3RSfs
  - image\_divc, [164](#)
- nppiDivC\_8u\_C3RSfs
  - image\_divc, [164](#)
- nppiDivC\_8u\_C4RSfs
  - image\_divc, [165](#)
- nppiDivC\_8u\_C4RSfs
  - image\_divc, [165](#)
- nppiDotProd\_16s64f\_AC4R

- image\_dot\_prod, [1805](#)
- nppiDotProd\_16s64f\_C1R
  - image\_dot\_prod, [1805](#)
- nppiDotProd\_16s64f\_C3R
  - image\_dot\_prod, [1806](#)
- nppiDotProd\_16s64f\_C4R
  - image\_dot\_prod, [1806](#)
- nppiDotProd\_16u64f\_AC4R
  - image\_dot\_prod, [1807](#)
- nppiDotProd\_16u64f\_C1R
  - image\_dot\_prod, [1807](#)
- nppiDotProd\_16u64f\_C3R
  - image\_dot\_prod, [1808](#)
- nppiDotProd\_16u64f\_C4R
  - image\_dot\_prod, [1808](#)
- nppiDotProd\_32f64f\_AC4R
  - image\_dot\_prod, [1808](#)
- nppiDotProd\_32f64f\_C1R
  - image\_dot\_prod, [1809](#)
- nppiDotProd\_32f64f\_C3R
  - image\_dot\_prod, [1809](#)
- nppiDotProd\_32f64f\_C4R
  - image\_dot\_prod, [1810](#)
- nppiDotProd\_32s64f\_AC4R
  - image\_dot\_prod, [1810](#)
- nppiDotProd\_32s64f\_C1R
  - image\_dot\_prod, [1811](#)
- nppiDotProd\_32s64f\_C3R
  - image\_dot\_prod, [1811](#)
- nppiDotProd\_32s64f\_C4R
  - image\_dot\_prod, [1811](#)
- nppiDotProd\_32u64f\_AC4R
  - image\_dot\_prod, [1812](#)
- nppiDotProd\_32u64f\_C1R
  - image\_dot\_prod, [1812](#)
- nppiDotProd\_32u64f\_C3R
  - image\_dot\_prod, [1813](#)
- nppiDotProd\_32u64f\_C4R
  - image\_dot\_prod, [1813](#)
- nppiDotProd\_8s64f\_AC4R
  - image\_dot\_prod, [1814](#)
- nppiDotProd\_8s64f\_C1R
  - image\_dot\_prod, [1814](#)
- nppiDotProd\_8s64f\_C3R
  - image\_dot\_prod, [1814](#)
- nppiDotProd\_8s64f\_C4R
  - image\_dot\_prod, [1815](#)
- nppiDotProd\_8u64f\_AC4R
  - image\_dot\_prod, [1815](#)
- nppiDotProd\_8u64f\_C1R
  - image\_dot\_prod, [1816](#)
- nppiDotProd\_8u64f\_C3R
  - image\_dot\_prod, [1816](#)
- nppiDotProd\_8u64f\_C4R
  - image\_dot\_prod, [1817](#)
- nppiDotProdGetBufferHostSize\_16s64f\_AC4R
  - image\_dot\_prod, [1817](#)
- nppiDotProdGetBufferHostSize\_16s64f\_C1R
  - image\_dot\_prod, [1817](#)
- nppiDotProdGetBufferHostSize\_16s64f\_C3R
  - image\_dot\_prod, [1818](#)
- nppiDotProdGetBufferHostSize\_16s64f\_C4R
  - image\_dot\_prod, [1818](#)
- nppiDotProdGetBufferHostSize\_16u64f\_AC4R
  - image\_dot\_prod, [1818](#)
- nppiDotProdGetBufferHostSize\_16u64f\_C1R
  - image\_dot\_prod, [1818](#)
- nppiDotProdGetBufferHostSize\_16u64f\_C3R
  - image\_dot\_prod, [1819](#)
- nppiDotProdGetBufferHostSize\_16u64f\_C4R
  - image\_dot\_prod, [1819](#)
- nppiDotProdGetBufferHostSize\_32f64f\_AC4R
  - image\_dot\_prod, [1819](#)
- nppiDotProdGetBufferHostSize\_32f64f\_C1R
  - image\_dot\_prod, [1820](#)
- nppiDotProdGetBufferHostSize\_32f64f\_C3R
  - image\_dot\_prod, [1820](#)
- nppiDotProdGetBufferHostSize\_32f64f\_C4R
  - image\_dot\_prod, [1820](#)
- nppiDotProdGetBufferHostSize\_32s64f\_AC4R
  - image\_dot\_prod, [1820](#)
- nppiDotProdGetBufferHostSize\_32s64f\_C1R
  - image\_dot\_prod, [1821](#)
- nppiDotProdGetBufferHostSize\_32s64f\_C3R
  - image\_dot\_prod, [1821](#)
- nppiDotProdGetBufferHostSize\_32s64f\_C4R
  - image\_dot\_prod, [1821](#)
- nppiDotProdGetBufferHostSize\_32u64f\_AC4R
  - image\_dot\_prod, [1822](#)
- nppiDotProdGetBufferHostSize\_32u64f\_C1R
  - image\_dot\_prod, [1822](#)
- nppiDotProdGetBufferHostSize\_32u64f\_C3R
  - image\_dot\_prod, [1822](#)
- nppiDotProdGetBufferHostSize\_32u64f\_C4R
  - image\_dot\_prod, [1822](#)
- nppiDotProdGetBufferHostSize\_8s64f\_AC4R
  - image\_dot\_prod, [1823](#)
- nppiDotProdGetBufferHostSize\_8s64f\_C1R
  - image\_dot\_prod, [1823](#)
- nppiDotProdGetBufferHostSize\_8s64f\_C3R
  - image\_dot\_prod, [1823](#)
- nppiDotProdGetBufferHostSize\_8s64f\_C4R
  - image\_dot\_prod, [1824](#)
- nppiDotProdGetBufferHostSize\_8u64f\_AC4R
  - image\_dot\_prod, [1824](#)
- nppiDotProdGetBufferHostSize\_8u64f\_C1R
  - image\_dot\_prod, [1824](#)
- nppiDotProdGetBufferHostSize\_8u64f\_C3R
  - image\_dot\_prod, [1824](#)

- image\_dot\_prod, [1824](#)
- npPiDotProdGetBufferHostSize\_8u64f\_C4R
  - image\_dot\_prod, [1825](#)
- npPiDup\_16s\_C1AC4R
  - image\_duplicate\_channel, [910](#)
- npPiDup\_16s\_C1C3R
  - image\_duplicate\_channel, [910](#)
- npPiDup\_16s\_C1C4R
  - image\_duplicate\_channel, [911](#)
- npPiDup\_16u\_C1AC4R
  - image\_duplicate\_channel, [911](#)
- npPiDup\_16u\_C1C3R
  - image\_duplicate\_channel, [911](#)
- npPiDup\_16u\_C1C4R
  - image\_duplicate\_channel, [912](#)
- npPiDup\_32f\_C1AC4R
  - image\_duplicate\_channel, [912](#)
- npPiDup\_32f\_C1C3R
  - image\_duplicate\_channel, [912](#)
- npPiDup\_32f\_C1C4R
  - image\_duplicate\_channel, [913](#)
- npPiDup\_32s\_C1AC4R
  - image\_duplicate\_channel, [913](#)
- npPiDup\_32s\_C1C3R
  - image\_duplicate\_channel, [913](#)
- npPiDup\_32s\_C1C4R
  - image\_duplicate\_channel, [914](#)
- npPiDup\_8u\_C1AC4R
  - image\_duplicate\_channel, [914](#)
- npPiDup\_8u\_C1C3R
  - image\_duplicate\_channel, [914](#)
- npPiDup\_8u\_C1C4R
  - image\_duplicate\_channel, [915](#)
- npPiErode3x3\_16u\_AC4R
  - image\_erode\_3x3, [1388](#)
- npPiErode3x3\_16u\_C1R
  - image\_erode\_3x3, [1388](#)
- npPiErode3x3\_16u\_C3R
  - image\_erode\_3x3, [1388](#)
- npPiErode3x3\_16u\_C4R
  - image\_erode\_3x3, [1389](#)
- npPiErode3x3\_32f\_AC4R
  - image\_erode\_3x3, [1389](#)
- npPiErode3x3\_32f\_C1R
  - image\_erode\_3x3, [1389](#)
- npPiErode3x3\_32f\_C3R
  - image\_erode\_3x3, [1390](#)
- npPiErode3x3\_32f\_C4R
  - image\_erode\_3x3, [1390](#)
- npPiErode3x3\_64f\_C1R
  - image\_erode\_3x3, [1390](#)
- npPiErode3x3\_8u\_AC4R
  - image\_erode\_3x3, [1391](#)
- npPiErode3x3\_8u\_C1R
  - image\_erode\_3x3, [1391](#)
- npPiErode3x3\_8u\_C3R
  - image\_erode\_3x3, [1391](#)
- npPiErode3x3\_8u\_C4R
  - image\_erode\_3x3, [1392](#)
- npPiErode\_16u\_AC4R
  - image\_erode, [1375](#)
- npPiErode\_16u\_C1R
  - image\_erode, [1375](#)
- npPiErode\_16u\_C3R
  - image\_erode, [1376](#)
- npPiErode\_16u\_C4R
  - image\_erode, [1376](#)
- npPiErode\_32f\_AC4R
  - image\_erode, [1376](#)
- npPiErode\_32f\_C1R
  - image\_erode, [1377](#)
- npPiErode\_32f\_C3R
  - image\_erode, [1377](#)
- npPiErode\_32f\_C4R
  - image\_erode, [1378](#)
- npPiErode\_8u\_AC4R
  - image\_erode, [1378](#)
- npPiErode\_8u\_C1R
  - image\_erode, [1379](#)
- npPiErode\_8u\_C3R
  - image\_erode, [1379](#)
- npPiErode\_8u\_C4R
  - image\_erode, [1379](#)
- npPiEvenLevelsHost\_32s
  - image\_histogrameven, [1856](#)
- npPiExp\_16s\_C1IRSfs
  - image\_exp, [364](#)
- npPiExp\_16s\_C1RSfs
  - image\_exp, [364](#)
- npPiExp\_16s\_C3IRSfs
  - image\_exp, [365](#)
- npPiExp\_16s\_C3RSfs
  - image\_exp, [365](#)
- npPiExp\_16u\_C1IRSfs
  - image\_exp, [365](#)
- npPiExp\_16u\_C1RSfs
  - image\_exp, [366](#)
- npPiExp\_16u\_C3IRSfs
  - image\_exp, [366](#)
- npPiExp\_16u\_C3RSfs
  - image\_exp, [366](#)
- npPiExp\_32f\_C1IR
  - image\_exp, [367](#)
- npPiExp\_32f\_C1R
  - image\_exp, [367](#)
- npPiExp\_32f\_C3IR
  - image\_exp, [367](#)
- npPiExp\_32f\_C3R

- image\_exp, 368
- nppiExp\_8u\_C1RSfs
  - image\_exp, 368
- nppiExp\_8u\_C1RSfs
  - image\_exp, 368
- nppiExp\_8u\_C3RSfs
  - image\_exp, 369
- nppiExp\_8u\_C3RSfs
  - image\_exp, 369
- nppiFilter32f\_16s\_AC4R
  - image\_convolution, 1066
- nppiFilter32f\_16s\_C1R
  - image\_convolution, 1066
- nppiFilter32f\_16s\_C3R
  - image\_convolution, 1066
- nppiFilter32f\_16s\_C4R
  - image\_convolution, 1067
- nppiFilter32f\_16u\_AC4R
  - image\_convolution, 1067
- nppiFilter32f\_16u\_C1R
  - image\_convolution, 1068
- nppiFilter32f\_16u\_C3R
  - image\_convolution, 1068
- nppiFilter32f\_16u\_C4R
  - image\_convolution, 1069
- nppiFilter32f\_32s\_AC4R
  - image\_convolution, 1069
- nppiFilter32f\_32s\_C1R
  - image\_convolution, 1070
- nppiFilter32f\_32s\_C3R
  - image\_convolution, 1070
- nppiFilter32f\_32s\_C4R
  - image\_convolution, 1071
- nppiFilter32f\_8s16s\_AC4R
  - image\_convolution, 1071
- nppiFilter32f\_8s16s\_C1R
  - image\_convolution, 1072
- nppiFilter32f\_8s16s\_C3R
  - image\_convolution, 1072
- nppiFilter32f\_8s16s\_C4R
  - image\_convolution, 1073
- nppiFilter32f\_8s\_AC4R
  - image\_convolution, 1073
- nppiFilter32f\_8s\_C1R
  - image\_convolution, 1074
- nppiFilter32f\_8s\_C3R
  - image\_convolution, 1074
- nppiFilter32f\_8s\_C4R
  - image\_convolution, 1075
- nppiFilter32f\_8u16s\_AC4R
  - image\_convolution, 1075
- nppiFilter32f\_8u16s\_C1R
  - image\_convolution, 1076
- nppiFilter32f\_8u16s\_C3R
  - image\_convolution, 1076
- nppiFilter32f\_8u16s\_C4R
  - image\_convolution, 1077
- nppiFilter32f\_8u\_AC4R
  - image\_convolution, 1077
- nppiFilter32f\_8u\_C1R
  - image\_convolution, 1078
- nppiFilter32f\_8u\_C3R
  - image\_convolution, 1078
- nppiFilter32f\_8u\_C4R
  - image\_convolution, 1079
- nppiFilter\_16s\_AC4R
  - image\_convolution, 1079
- nppiFilter\_16s\_C1R
  - image\_convolution, 1080
- nppiFilter\_16s\_C3R
  - image\_convolution, 1080
- nppiFilter\_16s\_C4R
  - image\_convolution, 1081
- nppiFilter\_16u\_AC4R
  - image\_convolution, 1081
- nppiFilter\_16u\_C1R
  - image\_convolution, 1082
- nppiFilter\_16u\_C3R
  - image\_convolution, 1082
- nppiFilter\_16u\_C4R
  - image\_convolution, 1083
- nppiFilter\_32f\_AC4R
  - image\_convolution, 1083
- nppiFilter\_32f\_C1R
  - image\_convolution, 1084
- nppiFilter\_32f\_C3R
  - image\_convolution, 1084
- nppiFilter\_32f\_C4R
  - image\_convolution, 1085
- nppiFilter\_64f\_C1R
  - image\_convolution, 1085
- nppiFilter\_8u\_AC4R
  - image\_convolution, 1086
- nppiFilter\_8u\_C1R
  - image\_convolution, 1086
- nppiFilter\_8u\_C3R
  - image\_convolution, 1087
- nppiFilter\_8u\_C4R
  - image\_convolution, 1087
- nppiFilterBorder32f\_16s\_AC4R
  - image\_convolution, 1088
- nppiFilterBorder32f\_16s\_C1R
  - image\_convolution, 1088
- nppiFilterBorder32f\_16s\_C3R
  - image\_convolution, 1089
- nppiFilterBorder32f\_16s\_C4R
  - image\_convolution, 1089
- nppiFilterBorder32f\_16u\_AC4R
  - image\_convolution, 1089

- image\_convolution, [1090](#)
- nppiFilterBorder32f\_16u\_C1R
  - image\_convolution, [1090](#)
- nppiFilterBorder32f\_16u\_C3R
  - image\_convolution, [1091](#)
- nppiFilterBorder32f\_16u\_C4R
  - image\_convolution, [1091](#)
- nppiFilterBorder32f\_32s\_AC4R
  - image\_convolution, [1092](#)
- nppiFilterBorder32f\_32s\_C1R
  - image\_convolution, [1092](#)
- nppiFilterBorder32f\_32s\_C3R
  - image\_convolution, [1093](#)
- nppiFilterBorder32f\_32s\_C4R
  - image\_convolution, [1093](#)
- nppiFilterBorder32f\_8s16s\_AC4R
  - image\_convolution, [1094](#)
- nppiFilterBorder32f\_8s16s\_C1R
  - image\_convolution, [1094](#)
- nppiFilterBorder32f\_8s16s\_C3R
  - image\_convolution, [1095](#)
- nppiFilterBorder32f\_8s16s\_C4R
  - image\_convolution, [1095](#)
- nppiFilterBorder32f\_8s\_AC4R
  - image\_convolution, [1096](#)
- nppiFilterBorder32f\_8s\_C1R
  - image\_convolution, [1096](#)
- nppiFilterBorder32f\_8s\_C3R
  - image\_convolution, [1097](#)
- nppiFilterBorder32f\_8s\_C4R
  - image\_convolution, [1097](#)
- nppiFilterBorder32f\_8u16s\_AC4R
  - image\_convolution, [1098](#)
- nppiFilterBorder32f\_8u16s\_C1R
  - image\_convolution, [1098](#)
- nppiFilterBorder32f\_8u16s\_C3R
  - image\_convolution, [1099](#)
- nppiFilterBorder32f\_8u16s\_C4R
  - image\_convolution, [1099](#)
- nppiFilterBorder32f\_8u\_AC4R
  - image\_convolution, [1100](#)
- nppiFilterBorder32f\_8u\_C1R
  - image\_convolution, [1100](#)
- nppiFilterBorder32f\_8u\_C3R
  - image\_convolution, [1101](#)
- nppiFilterBorder32f\_8u\_C4R
  - image\_convolution, [1101](#)
- nppiFilterBorder\_16s\_AC4R
  - image\_convolution, [1102](#)
- nppiFilterBorder\_16s\_C1R
  - image\_convolution, [1103](#)
- nppiFilterBorder\_16s\_C3R
  - image\_convolution, [1103](#)
- nppiFilterBorder\_16s\_C4R
  - image\_convolution, [1104](#)
- nppiFilterBorder\_16u\_AC4R
  - image\_convolution, [1104](#)
- nppiFilterBorder\_16u\_C1R
  - image\_convolution, [1105](#)
- nppiFilterBorder\_16u\_C3R
  - image\_convolution, [1106](#)
- nppiFilterBorder\_16u\_C4R
  - image\_convolution, [1106](#)
- nppiFilterBorder\_32f\_AC4R
  - image\_convolution, [1107](#)
- nppiFilterBorder\_32f\_C1R
  - image\_convolution, [1107](#)
- nppiFilterBorder\_32f\_C3R
  - image\_convolution, [1108](#)
- nppiFilterBorder\_32f\_C4R
  - image\_convolution, [1108](#)
- nppiFilterBorder\_8u\_AC4R
  - image\_convolution, [1109](#)
- nppiFilterBorder\_8u\_C1R
  - image\_convolution, [1110](#)
- nppiFilterBorder\_8u\_C3R
  - image\_convolution, [1110](#)
- nppiFilterBorder\_8u\_C4R
  - image\_convolution, [1111](#)
- nppiFilterBox\_16s\_AC4R
  - image\_2D\_fixed\_linear\_filters, [1113](#)
- nppiFilterBox\_16s\_C1R
  - image\_2D\_fixed\_linear\_filters, [1113](#)
- nppiFilterBox\_16s\_C3R
  - image\_2D\_fixed\_linear\_filters, [1114](#)
- nppiFilterBox\_16s\_C4R
  - image\_2D\_fixed\_linear\_filters, [1114](#)
- nppiFilterBox\_16u\_AC4R
  - image\_2D\_fixed\_linear\_filters, [1115](#)
- nppiFilterBox\_16u\_C1R
  - image\_2D\_fixed\_linear\_filters, [1115](#)
- nppiFilterBox\_16u\_C3R
  - image\_2D\_fixed\_linear\_filters, [1115](#)
- nppiFilterBox\_16u\_C4R
  - image\_2D\_fixed\_linear\_filters, [1116](#)
- nppiFilterBox\_32f\_AC4R
  - image\_2D\_fixed\_linear\_filters, [1116](#)
- nppiFilterBox\_32f\_C1R
  - image\_2D\_fixed\_linear\_filters, [1117](#)
- nppiFilterBox\_32f\_C3R
  - image\_2D\_fixed\_linear\_filters, [1117](#)
- nppiFilterBox\_32f\_C4R
  - image\_2D\_fixed\_linear\_filters, [1117](#)
- nppiFilterBox\_64f\_C1R
  - image\_2D\_fixed\_linear\_filters, [1118](#)
- nppiFilterBox\_8u\_AC4R
  - image\_2D\_fixed\_linear\_filters, [1118](#)
- nppiFilterBox\_8u\_C1R

- image\_2D\_fixed\_linear\_filters, [1119](#)
- npippiFilterBox\_8u\_C3R
  - image\_2D\_fixed\_linear\_filters, [1119](#)
- npippiFilterBox\_8u\_C4R
  - image\_2D\_fixed\_linear\_filters, [1119](#)
- npippiFilterColumn32f\_16s\_AC4R
  - image\_1D\_linear\_filter, [999](#)
- npippiFilterColumn32f\_16s\_C1R
  - image\_1D\_linear\_filter, [1000](#)
- npippiFilterColumn32f\_16s\_C3R
  - image\_1D\_linear\_filter, [1000](#)
- npippiFilterColumn32f\_16s\_C4R
  - image\_1D\_linear\_filter, [1000](#)
- npippiFilterColumn32f\_16u\_AC4R
  - image\_1D\_linear\_filter, [1001](#)
- npippiFilterColumn32f\_16u\_C1R
  - image\_1D\_linear\_filter, [1001](#)
- npippiFilterColumn32f\_16u\_C3R
  - image\_1D\_linear\_filter, [1002](#)
- npippiFilterColumn32f\_16u\_C4R
  - image\_1D\_linear\_filter, [1002](#)
- npippiFilterColumn32f\_8u\_AC4R
  - image\_1D\_linear\_filter, [1003](#)
- npippiFilterColumn32f\_8u\_C1R
  - image\_1D\_linear\_filter, [1003](#)
- npippiFilterColumn32f\_8u\_C3R
  - image\_1D\_linear\_filter, [1004](#)
- npippiFilterColumn32f\_8u\_C4R
  - image\_1D\_linear\_filter, [1004](#)
- npippiFilterColumn\_16s\_AC4R
  - image\_1D\_linear\_filter, [1005](#)
- npippiFilterColumn\_16s\_C1R
  - image\_1D\_linear\_filter, [1005](#)
- npippiFilterColumn\_16s\_C3R
  - image\_1D\_linear\_filter, [1006](#)
- npippiFilterColumn\_16s\_C4R
  - image\_1D\_linear\_filter, [1006](#)
- npippiFilterColumn\_16u\_AC4R
  - image\_1D\_linear\_filter, [1007](#)
- npippiFilterColumn\_16u\_C1R
  - image\_1D\_linear\_filter, [1007](#)
- npippiFilterColumn\_16u\_C3R
  - image\_1D\_linear\_filter, [1008](#)
- npippiFilterColumn\_16u\_C4R
  - image\_1D\_linear\_filter, [1008](#)
- npippiFilterColumn\_32f\_AC4R
  - image\_1D\_linear\_filter, [1009](#)
- npippiFilterColumn\_32f\_C1R
  - image\_1D\_linear\_filter, [1009](#)
- npippiFilterColumn\_32f\_C3R
  - image\_1D\_linear\_filter, [1010](#)
- npippiFilterColumn\_32f\_C4R
  - image\_1D\_linear\_filter, [1010](#)
- npippiFilterColumn\_64f\_C1R
  - image\_1D\_linear\_filter, [1011](#)
- npippiFilterColumn\_8u\_AC4R
  - image\_1D\_linear\_filter, [1011](#)
- npippiFilterColumn\_8u\_C1R
  - image\_1D\_linear\_filter, [1012](#)
- npippiFilterColumn\_8u\_C3R
  - image\_1D\_linear\_filter, [1012](#)
- npippiFilterColumn\_8u\_C4R
  - image\_1D\_linear\_filter, [1013](#)
- npippiFilterGauss\_16s\_AC4R
  - image\_filtering\_functions, [951](#)
- npippiFilterGauss\_16s\_C1R
  - image\_filtering\_functions, [952](#)
- npippiFilterGauss\_16s\_C3R
  - image\_filtering\_functions, [952](#)
- npippiFilterGauss\_16s\_C4R
  - image\_filtering\_functions, [952](#)
- npippiFilterGauss\_16u\_AC4R
  - image\_filtering\_functions, [953](#)
- npippiFilterGauss\_16u\_C1R
  - image\_filtering\_functions, [953](#)
- npippiFilterGauss\_16u\_C3R
  - image\_filtering\_functions, [953](#)
- npippiFilterGauss\_16u\_C4R
  - image\_filtering\_functions, [954](#)
- npippiFilterGauss\_32f\_AC4R
  - image\_filtering\_functions, [954](#)
- npippiFilterGauss\_32f\_C1R
  - image\_filtering\_functions, [954](#)
- npippiFilterGauss\_32f\_C3R
  - image\_filtering\_functions, [955](#)
- npippiFilterGauss\_32f\_C4R
  - image\_filtering\_functions, [955](#)
- npippiFilterGauss\_8u\_AC4R
  - image\_filtering\_functions, [955](#)
- npippiFilterGauss\_8u\_C1R
  - image\_filtering\_functions, [956](#)
- npippiFilterGauss\_8u\_C3R
  - image\_filtering\_functions, [956](#)
- npippiFilterGauss\_8u\_C4R
  - image\_filtering\_functions, [956](#)
- npippiFilterHighPass\_16s\_AC4R
  - image\_filtering\_functions, [957](#)
- npippiFilterHighPass\_16s\_C1R
  - image\_filtering\_functions, [957](#)
- npippiFilterHighPass\_16s\_C3R
  - image\_filtering\_functions, [957](#)
- npippiFilterHighPass\_16s\_C4R
  - image\_filtering\_functions, [958](#)
- npippiFilterHighPass\_16u\_AC4R
  - image\_filtering\_functions, [958](#)
- npippiFilterHighPass\_16u\_C1R
  - image\_filtering\_functions, [958](#)
- npippiFilterHighPass\_16u\_C3R



- image\_filtering\_functions, 959
- nppiFilterHighPass\_16u\_C4R
  - image\_filtering\_functions, 959
- nppiFilterHighPass\_32f\_AC4R
  - image\_filtering\_functions, 959
- nppiFilterHighPass\_32f\_C1R
  - image\_filtering\_functions, 960
- nppiFilterHighPass\_32f\_C3R
  - image\_filtering\_functions, 960
- nppiFilterHighPass\_32f\_C4R
  - image\_filtering\_functions, 960
- nppiFilterHighPass\_8u\_AC4R
  - image\_filtering\_functions, 961
- nppiFilterHighPass\_8u\_C1R
  - image\_filtering\_functions, 961
- nppiFilterHighPass\_8u\_C3R
  - image\_filtering\_functions, 961
- nppiFilterHighPass\_8u\_C4R
  - image\_filtering\_functions, 962
- nppiFilterLaplace\_16s\_AC4R
  - image\_filtering\_functions, 962
- nppiFilterLaplace\_16s\_C1R
  - image\_filtering\_functions, 962
- nppiFilterLaplace\_16s\_C3R
  - image\_filtering\_functions, 963
- nppiFilterLaplace\_16s\_C4R
  - image\_filtering\_functions, 963
- nppiFilterLaplace\_32f\_AC4R
  - image\_filtering\_functions, 963
- nppiFilterLaplace\_32f\_C1R
  - image\_filtering\_functions, 964
- nppiFilterLaplace\_32f\_C3R
  - image\_filtering\_functions, 964
- nppiFilterLaplace\_32f\_C4R
  - image\_filtering\_functions, 964
- nppiFilterLaplace\_8s16s\_C1R
  - image\_filtering\_functions, 965
- nppiFilterLaplace\_8u16s\_C1R
  - image\_filtering\_functions, 965
- nppiFilterLaplace\_8u\_AC4R
  - image\_filtering\_functions, 965
- nppiFilterLaplace\_8u\_C1R
  - image\_filtering\_functions, 966
- nppiFilterLaplace\_8u\_C3R
  - image\_filtering\_functions, 966
- nppiFilterLaplace\_8u\_C4R
  - image\_filtering\_functions, 966
- nppiFilterLowPass\_16s\_AC4R
  - image\_filtering\_functions, 967
- nppiFilterLowPass\_16s\_C1R
  - image\_filtering\_functions, 967
- nppiFilterLowPass\_16s\_C3R
  - image\_filtering\_functions, 967
- nppiFilterLowPass\_16s\_C4R
  - image\_filtering\_functions, 968
- nppiFilterLowPass\_16u\_AC4R
  - image\_filtering\_functions, 968
- nppiFilterLowPass\_16u\_C1R
  - image\_filtering\_functions, 968
- nppiFilterLowPass\_16u\_C3R
  - image\_filtering\_functions, 969
- nppiFilterLowPass\_16u\_C4R
  - image\_filtering\_functions, 969
- nppiFilterLowPass\_32f\_AC4R
  - image\_filtering\_functions, 969
- nppiFilterLowPass\_32f\_C1R
  - image\_filtering\_functions, 970
- nppiFilterLowPass\_32f\_C3R
  - image\_filtering\_functions, 970
- nppiFilterLowPass\_32f\_C4R
  - image\_filtering\_functions, 970
- nppiFilterLowPass\_8u\_AC4R
  - image\_filtering\_functions, 971
- nppiFilterLowPass\_8u\_C1R
  - image\_filtering\_functions, 971
- nppiFilterLowPass\_8u\_C3R
  - image\_filtering\_functions, 971
- nppiFilterLowPass\_8u\_C4R
  - image\_filtering\_functions, 972
- nppiFilterMax\_16s\_AC4R
  - image\_rank\_filters, 1126
- nppiFilterMax\_16s\_C1R
  - image\_rank\_filters, 1127
- nppiFilterMax\_16s\_C3R
  - image\_rank\_filters, 1127
- nppiFilterMax\_16s\_C4R
  - image\_rank\_filters, 1127
- nppiFilterMax\_16u\_AC4R
  - image\_rank\_filters, 1128
- nppiFilterMax\_16u\_C1R
  - image\_rank\_filters, 1128
- nppiFilterMax\_16u\_C3R
  - image\_rank\_filters, 1128
- nppiFilterMax\_16u\_C4R
  - image\_rank\_filters, 1129
- nppiFilterMax\_32f\_AC4R
  - image\_rank\_filters, 1129
- nppiFilterMax\_32f\_C1R
  - image\_rank\_filters, 1130
- nppiFilterMax\_32f\_C3R
  - image\_rank\_filters, 1130
- nppiFilterMax\_32f\_C4R
  - image\_rank\_filters, 1130
- nppiFilterMax\_8u\_AC4R
  - image\_rank\_filters, 1131
- nppiFilterMax\_8u\_C1R
  - image\_rank\_filters, 1131
- nppiFilterMax\_8u\_C3R

- image\_rank\_filters, [1132](#)
- npippiFilterMax\_8u\_C4R
  - image\_rank\_filters, [1132](#)
- npippiFilterMedian\_16s\_AC4R
  - image\_rank\_filters, [1132](#)
- npippiFilterMedian\_16s\_C1R
  - image\_rank\_filters, [1133](#)
- npippiFilterMedian\_16s\_C3R
  - image\_rank\_filters, [1133](#)
- npippiFilterMedian\_16s\_C4R
  - image\_rank\_filters, [1134](#)
- npippiFilterMedian\_16u\_AC4R
  - image\_rank\_filters, [1134](#)
- npippiFilterMedian\_16u\_C1R
  - image\_rank\_filters, [1135](#)
- npippiFilterMedian\_16u\_C3R
  - image\_rank\_filters, [1135](#)
- npippiFilterMedian\_16u\_C4R
  - image\_rank\_filters, [1135](#)
- npippiFilterMedian\_32f\_AC4R
  - image\_rank\_filters, [1136](#)
- npippiFilterMedian\_32f\_C1R
  - image\_rank\_filters, [1136](#)
- npippiFilterMedian\_32f\_C3R
  - image\_rank\_filters, [1137](#)
- npippiFilterMedian\_32f\_C4R
  - image\_rank\_filters, [1137](#)
- npippiFilterMedian\_8u\_AC4R
  - image\_rank\_filters, [1138](#)
- npippiFilterMedian\_8u\_C1R
  - image\_rank\_filters, [1138](#)
- npippiFilterMedian\_8u\_C3R
  - image\_rank\_filters, [1138](#)
- npippiFilterMedian\_8u\_C4R
  - image\_rank\_filters, [1139](#)
- npippiFilterMedianGetBufferSize\_16s\_AC4R
  - image\_rank\_filters, [1139](#)
- npippiFilterMedianGetBufferSize\_16s\_C1R
  - image\_rank\_filters, [1140](#)
- npippiFilterMedianGetBufferSize\_16s\_C3R
  - image\_rank\_filters, [1140](#)
- npippiFilterMedianGetBufferSize\_16s\_C4R
  - image\_rank\_filters, [1140](#)
- npippiFilterMedianGetBufferSize\_16u\_AC4R
  - image\_rank\_filters, [1140](#)
- npippiFilterMedianGetBufferSize\_16u\_C1R
  - image\_rank\_filters, [1141](#)
- npippiFilterMedianGetBufferSize\_16u\_C3R
  - image\_rank\_filters, [1141](#)
- npippiFilterMedianGetBufferSize\_16u\_C4R
  - image\_rank\_filters, [1141](#)
- npippiFilterMedianGetBufferSize\_32f\_AC4R
  - image\_rank\_filters, [1142](#)
- npippiFilterMedianGetBufferSize\_32f\_C1R
  - image\_rank\_filters, [1142](#)
- npippiFilterMedianGetBufferSize\_32f\_C3R
  - image\_rank\_filters, [1142](#)
- npippiFilterMedianGetBufferSize\_32f\_C4R
  - image\_rank\_filters, [1142](#)
- npippiFilterMedianGetBufferSize\_8u\_AC4R
  - image\_rank\_filters, [1143](#)
- npippiFilterMedianGetBufferSize\_8u\_C1R
  - image\_rank\_filters, [1143](#)
- npippiFilterMedianGetBufferSize\_8u\_C3R
  - image\_rank\_filters, [1143](#)
- npippiFilterMedianGetBufferSize\_8u\_C4R
  - image\_rank\_filters, [1144](#)
- npippiFilterMin\_16s\_AC4R
  - image\_rank\_filters, [1144](#)
- npippiFilterMin\_16s\_C1R
  - image\_rank\_filters, [1144](#)
- npippiFilterMin\_16s\_C3R
  - image\_rank\_filters, [1145](#)
- npippiFilterMin\_16s\_C4R
  - image\_rank\_filters, [1145](#)
- npippiFilterMin\_16u\_AC4R
  - image\_rank\_filters, [1146](#)
- npippiFilterMin\_16u\_C1R
  - image\_rank\_filters, [1146](#)
- npippiFilterMin\_16u\_C3R
  - image\_rank\_filters, [1146](#)
- npippiFilterMin\_16u\_C4R
  - image\_rank\_filters, [1147](#)
- npippiFilterMin\_32f\_AC4R
  - image\_rank\_filters, [1147](#)
- npippiFilterMin\_32f\_C1R
  - image\_rank\_filters, [1148](#)
- npippiFilterMin\_32f\_C3R
  - image\_rank\_filters, [1148](#)
- npippiFilterMin\_32f\_C4R
  - image\_rank\_filters, [1148](#)
- npippiFilterMin\_8u\_AC4R
  - image\_rank\_filters, [1149](#)
- npippiFilterMin\_8u\_C1R
  - image\_rank\_filters, [1149](#)
- npippiFilterMin\_8u\_C3R
  - image\_rank\_filters, [1150](#)
- npippiFilterMin\_8u\_C4R
  - image\_rank\_filters, [1150](#)
- npippiFilterPrewittHoriz\_16s\_AC4R
  - fixed\_filters, [1158](#)
- npippiFilterPrewittHoriz\_16s\_C1R
  - fixed\_filters, [1158](#)
- npippiFilterPrewittHoriz\_16s\_C3R
  - fixed\_filters, [1159](#)
- npippiFilterPrewittHoriz\_16s\_C4R
  - fixed\_filters, [1159](#)
- npippiFilterPrewittHoriz\_32f\_AC4R



- fixed\_filters, [1159](#)
- [nppiFilterPrewittHoriz\\_32f\\_C1R](#)
  - fixed\_filters, [1160](#)
- [nppiFilterPrewittHoriz\\_32f\\_C3R](#)
  - fixed\_filters, [1160](#)
- [nppiFilterPrewittHoriz\\_32f\\_C4R](#)
  - fixed\_filters, [1160](#)
- [nppiFilterPrewittHoriz\\_8u\\_AC4R](#)
  - fixed\_filters, [1161](#)
- [nppiFilterPrewittHoriz\\_8u\\_C1R](#)
  - fixed\_filters, [1161](#)
- [nppiFilterPrewittHoriz\\_8u\\_C3R](#)
  - fixed\_filters, [1161](#)
- [nppiFilterPrewittHoriz\\_8u\\_C4R](#)
  - fixed\_filters, [1162](#)
- [nppiFilterPrewittVert\\_16s\\_AC4R](#)
  - fixed\_filters, [1162](#)
- [nppiFilterPrewittVert\\_16s\\_C1R](#)
  - fixed\_filters, [1162](#)
- [nppiFilterPrewittVert\\_16s\\_C3R](#)
  - fixed\_filters, [1163](#)
- [nppiFilterPrewittVert\\_16s\\_C4R](#)
  - fixed\_filters, [1163](#)
- [nppiFilterPrewittVert\\_32f\\_AC4R](#)
  - fixed\_filters, [1163](#)
- [nppiFilterPrewittVert\\_32f\\_C1R](#)
  - fixed\_filters, [1164](#)
- [nppiFilterPrewittVert\\_32f\\_C3R](#)
  - fixed\_filters, [1164](#)
- [nppiFilterPrewittVert\\_32f\\_C4R](#)
  - fixed\_filters, [1164](#)
- [nppiFilterPrewittVert\\_8u\\_AC4R](#)
  - fixed\_filters, [1165](#)
- [nppiFilterPrewittVert\\_8u\\_C1R](#)
  - fixed\_filters, [1165](#)
- [nppiFilterPrewittVert\\_8u\\_C3R](#)
  - fixed\_filters, [1165](#)
- [nppiFilterPrewittVert\\_8u\\_C4R](#)
  - fixed\_filters, [1166](#)
- [nppiFilterRobertsDown\\_16s\\_AC4R](#)
  - image\_filtering\_functions, [972](#)
- [nppiFilterRobertsDown\\_16s\\_C1R](#)
  - image\_filtering\_functions, [972](#)
- [nppiFilterRobertsDown\\_16s\\_C3R](#)
  - image\_filtering\_functions, [973](#)
- [nppiFilterRobertsDown\\_16s\\_C4R](#)
  - image\_filtering\_functions, [973](#)
- [nppiFilterRobertsDown\\_32f\\_AC4R](#)
  - image\_filtering\_functions, [973](#)
- [nppiFilterRobertsDown\\_32f\\_C1R](#)
  - image\_filtering\_functions, [974](#)
- [nppiFilterRobertsDown\\_32f\\_C3R](#)
  - image\_filtering\_functions, [974](#)
- [nppiFilterRobertsDown\\_32f\\_C4R](#)
  - image\_filtering\_functions, [974](#)
- [nppiFilterRobertsDown\\_8u\\_AC4R](#)
  - image\_filtering\_functions, [975](#)
- [nppiFilterRobertsDown\\_8u\\_C1R](#)
  - image\_filtering\_functions, [975](#)
- [nppiFilterRobertsDown\\_8u\\_C3R](#)
  - image\_filtering\_functions, [975](#)
- [nppiFilterRobertsDown\\_8u\\_C4R](#)
  - image\_filtering\_functions, [976](#)
- [nppiFilterRobertsUp\\_16s\\_AC4R](#)
  - image\_filtering\_functions, [976](#)
- [nppiFilterRobertsUp\\_16s\\_C1R](#)
  - image\_filtering\_functions, [976](#)
- [nppiFilterRobertsUp\\_16s\\_C3R](#)
  - image\_filtering\_functions, [977](#)
- [nppiFilterRobertsUp\\_16s\\_C4R](#)
  - image\_filtering\_functions, [977](#)
- [nppiFilterRobertsUp\\_32f\\_AC4R](#)
  - image\_filtering\_functions, [977](#)
- [nppiFilterRobertsUp\\_32f\\_C1R](#)
  - image\_filtering\_functions, [978](#)
- [nppiFilterRobertsUp\\_32f\\_C3R](#)
  - image\_filtering\_functions, [978](#)
- [nppiFilterRobertsUp\\_32f\\_C4R](#)
  - image\_filtering\_functions, [978](#)
- [nppiFilterRobertsUp\\_8u\\_AC4R](#)
  - image\_filtering\_functions, [979](#)
- [nppiFilterRobertsUp\\_8u\\_C1R](#)
  - image\_filtering\_functions, [979](#)
- [nppiFilterRobertsUp\\_8u\\_C3R](#)
  - image\_filtering\_functions, [979](#)
- [nppiFilterRobertsUp\\_8u\\_C4R](#)
  - image\_filtering\_functions, [980](#)
- [nppiFilterRow32f\\_16s\\_AC4R](#)
  - image\_1D\_linear\_filter, [1013](#)
- [nppiFilterRow32f\\_16s\\_C1R](#)
  - image\_1D\_linear\_filter, [1014](#)
- [nppiFilterRow32f\\_16s\\_C3R](#)
  - image\_1D\_linear\_filter, [1014](#)
- [nppiFilterRow32f\\_16s\\_C4R](#)
  - image\_1D\_linear\_filter, [1015](#)
- [nppiFilterRow32f\\_16u\\_AC4R](#)
  - image\_1D\_linear\_filter, [1015](#)
- [nppiFilterRow32f\\_16u\\_C1R](#)
  - image\_1D\_linear\_filter, [1016](#)
- [nppiFilterRow32f\\_16u\\_C3R](#)
  - image\_1D\_linear\_filter, [1016](#)
- [nppiFilterRow32f\\_16u\\_C4R](#)
  - image\_1D\_linear\_filter, [1017](#)
- [nppiFilterRow32f\\_8u\\_AC4R](#)
  - image\_1D\_linear\_filter, [1017](#)
- [nppiFilterRow32f\\_8u\\_C1R](#)
  - image\_1D\_linear\_filter, [1018](#)
- [nppiFilterRow32f\\_8u\\_C3R](#)

image\_1D\_linear\_filter, [1018](#)  
 nppiFilterRow32f\_8u\_C4R  
     image\_1D\_linear\_filter, [1019](#)  
 nppiFilterRow\_16s\_AC4R  
     image\_1D\_linear\_filter, [1019](#)  
 nppiFilterRow\_16s\_C1R  
     image\_1D\_linear\_filter, [1020](#)  
 nppiFilterRow\_16s\_C3R  
     image\_1D\_linear\_filter, [1020](#)  
 nppiFilterRow\_16s\_C4R  
     image\_1D\_linear\_filter, [1021](#)  
 nppiFilterRow\_16u\_AC4R  
     image\_1D\_linear\_filter, [1021](#)  
 nppiFilterRow\_16u\_C1R  
     image\_1D\_linear\_filter, [1022](#)  
 nppiFilterRow\_16u\_C3R  
     image\_1D\_linear\_filter, [1022](#)  
 nppiFilterRow\_16u\_C4R  
     image\_1D\_linear\_filter, [1023](#)  
 nppiFilterRow\_32f\_AC4R  
     image\_1D\_linear\_filter, [1023](#)  
 nppiFilterRow\_32f\_C1R  
     image\_1D\_linear\_filter, [1024](#)  
 nppiFilterRow\_32f\_C3R  
     image\_1D\_linear\_filter, [1024](#)  
 nppiFilterRow\_32f\_C4R  
     image\_1D\_linear\_filter, [1025](#)  
 nppiFilterRow\_64f\_C1R  
     image\_1D\_linear\_filter, [1025](#)  
 nppiFilterRow\_8u\_AC4R  
     image\_1D\_linear\_filter, [1026](#)  
 nppiFilterRow\_8u\_C1R  
     image\_1D\_linear\_filter, [1026](#)  
 nppiFilterRow\_8u\_C3R  
     image\_1D\_linear\_filter, [1027](#)  
 nppiFilterRow\_8u\_C4R  
     image\_1D\_linear\_filter, [1027](#)  
 nppiFilterScharrHoriz\_32f\_C1R  
     fixed\_filters, [1166](#)  
 nppiFilterScharrHoriz\_8s16s\_C1R  
     fixed\_filters, [1166](#)  
 nppiFilterScharrHoriz\_8u16s\_C1R  
     fixed\_filters, [1167](#)  
 nppiFilterScharrHorizBorder\_32f\_C1R  
     fixed\_filters, [1167](#)  
 nppiFilterScharrHorizBorder\_8s16s\_C1R  
     fixed\_filters, [1168](#)  
 nppiFilterScharrHorizBorder\_8u16s\_C1R  
     fixed\_filters, [1168](#)  
 nppiFilterScharrVert\_32f\_C1R  
     fixed\_filters, [1168](#)  
 nppiFilterScharrVert\_8s16s\_C1R  
     fixed\_filters, [1169](#)  
 nppiFilterScharrVert\_8u16s\_C1R

    fixed\_filters, [1169](#)  
 nppiFilterScharrVertBorder\_32f\_C1R  
     fixed\_filters, [1169](#)  
 nppiFilterScharrVertBorder\_8s16s\_C1R  
     fixed\_filters, [1170](#)  
 nppiFilterScharrVertBorder\_8u16s\_C1R  
     fixed\_filters, [1170](#)  
 nppiFilterSharpen\_16s\_AC4R  
     image\_filtering\_functions, [980](#)  
 nppiFilterSharpen\_16s\_C1R  
     image\_filtering\_functions, [980](#)  
 nppiFilterSharpen\_16s\_C3R  
     image\_filtering\_functions, [981](#)  
 nppiFilterSharpen\_16s\_C4R  
     image\_filtering\_functions, [981](#)  
 nppiFilterSharpen\_16u\_AC4R  
     image\_filtering\_functions, [981](#)  
 nppiFilterSharpen\_16u\_C1R  
     image\_filtering\_functions, [982](#)  
 nppiFilterSharpen\_16u\_C3R  
     image\_filtering\_functions, [982](#)  
 nppiFilterSharpen\_16u\_C4R  
     image\_filtering\_functions, [982](#)  
 nppiFilterSharpen\_32f\_AC4R  
     image\_filtering\_functions, [983](#)  
 nppiFilterSharpen\_32f\_C1R  
     image\_filtering\_functions, [983](#)  
 nppiFilterSharpen\_32f\_C3R  
     image\_filtering\_functions, [983](#)  
 nppiFilterSharpen\_32f\_C4R  
     image\_filtering\_functions, [984](#)  
 nppiFilterSharpen\_8u\_AC4R  
     image\_filtering\_functions, [984](#)  
 nppiFilterSharpen\_8u\_C1R  
     image\_filtering\_functions, [984](#)  
 nppiFilterSharpen\_8u\_C3R  
     image\_filtering\_functions, [985](#)  
 nppiFilterSharpen\_8u\_C4R  
     image\_filtering\_functions, [985](#)  
 nppiFilterSobelCross\_32f\_C1R  
     image\_1D\_linear\_filter, [1028](#)  
 nppiFilterSobelCross\_8s16s\_C1R  
     image\_1D\_linear\_filter, [1028](#)  
 nppiFilterSobelCross\_8u16s\_C1R  
     image\_1D\_linear\_filter, [1029](#)  
 nppiFilterSobelCrossBorder\_32f\_C1R  
     image\_filtering\_functions, [985](#)  
 nppiFilterSobelCrossBorder\_8s16s\_C1R  
     image\_filtering\_functions, [986](#)  
 nppiFilterSobelCrossBorder\_8u16s\_C1R  
     image\_filtering\_functions, [986](#)  
 nppiFilterSobelHoriz\_16s\_AC4R  
     fixed\_filters, [1171](#)  
 nppiFilterSobelHoriz\_16s\_C1R

- fixed\_filters, [1171](#)
- nppiFilterSobelHoriz\_16s\_C3R
  - fixed\_filters, [1171](#)
- nppiFilterSobelHoriz\_16s\_C4R
  - fixed\_filters, [1172](#)
- nppiFilterSobelHoriz\_32f\_AC4R
  - fixed\_filters, [1172](#)
- nppiFilterSobelHoriz\_32f\_C1R
  - fixed\_filters, [1172](#)
- nppiFilterSobelHoriz\_32f\_C3R
  - fixed\_filters, [1173](#)
- nppiFilterSobelHoriz\_32f\_C4R
  - fixed\_filters, [1173](#)
- nppiFilterSobelHoriz\_8s16s\_C1R
  - fixed\_filters, [1173](#)
- nppiFilterSobelHoriz\_8u16s\_C1R
  - fixed\_filters, [1174](#)
- nppiFilterSobelHoriz\_8u\_AC4R
  - fixed\_filters, [1174](#)
- nppiFilterSobelHoriz\_8u\_C1R
  - fixed\_filters, [1174](#)
- nppiFilterSobelHoriz\_8u\_C3R
  - fixed\_filters, [1175](#)
- nppiFilterSobelHoriz\_8u\_C4R
  - fixed\_filters, [1175](#)
- nppiFilterSobelHorizBorder\_16s\_AC4R
  - image\_1D\_linear\_filter, [1029](#)
- nppiFilterSobelHorizBorder\_16s\_C1R
  - image\_1D\_linear\_filter, [1029](#)
- nppiFilterSobelHorizBorder\_16s\_C3R
  - image\_1D\_linear\_filter, [1030](#)
- nppiFilterSobelHorizBorder\_16s\_C4R
  - image\_1D\_linear\_filter, [1030](#)
- nppiFilterSobelHorizBorder\_32f\_AC4R
  - image\_1D\_linear\_filter, [1031](#)
- nppiFilterSobelHorizBorder\_32f\_C1R
  - image\_1D\_linear\_filter, [1031](#)
- nppiFilterSobelHorizBorder\_32f\_C3R
  - image\_1D\_linear\_filter, [1032](#)
- nppiFilterSobelHorizBorder\_32f\_C4R
  - image\_1D\_linear\_filter, [1032](#)
- nppiFilterSobelHorizBorder\_8s16s\_C1R
  - image\_1D\_linear\_filter, [1032](#)
- nppiFilterSobelHorizBorder\_8u16s\_C1R
  - image\_1D\_linear\_filter, [1033](#)
- nppiFilterSobelHorizBorder\_8u\_AC4R
  - image\_1D\_linear\_filter, [1033](#)
- nppiFilterSobelHorizBorder\_8u\_C1R
  - image\_1D\_linear\_filter, [1034](#)
- nppiFilterSobelHorizBorder\_8u\_C3R
  - image\_1D\_linear\_filter, [1034](#)
- nppiFilterSobelHorizBorder\_8u\_C4R
  - image\_1D\_linear\_filter, [1035](#)
- nppiFilterSobelHorizMask\_32f\_C1R
  - fixed\_filters, [1175](#)
- nppiFilterSobelHorizMaskBorder\_32f\_C1R
  - image\_1D\_linear\_filter, [1035](#)
- nppiFilterSobelHorizSecond\_32f\_C1R
  - fixed\_filters, [1176](#)
- nppiFilterSobelHorizSecond\_8s16s\_C1R
  - fixed\_filters, [1176](#)
- nppiFilterSobelHorizSecond\_8u16s\_C1R
  - fixed\_filters, [1177](#)
- nppiFilterSobelHorizSecondBorder\_32f\_C1R
  - image\_1D\_linear\_filter, [1036](#)
- nppiFilterSobelHorizSecondBorder\_8s16s\_C1R
  - image\_1D\_linear\_filter, [1036](#)
- nppiFilterSobelHorizSecondBorder\_8u16s\_C1R
  - image\_1D\_linear\_filter, [1037](#)
- nppiFilterSobelVert\_16s\_AC4R
  - fixed\_filters, [1177](#)
- nppiFilterSobelVert\_16s\_C1R
  - fixed\_filters, [1177](#)
- nppiFilterSobelVert\_16s\_C3R
  - fixed\_filters, [1178](#)
- nppiFilterSobelVert\_16s\_C4R
  - fixed\_filters, [1178](#)
- nppiFilterSobelVert\_32f\_AC4R
  - fixed\_filters, [1178](#)
- nppiFilterSobelVert\_32f\_C1R
  - fixed\_filters, [1179](#)
- nppiFilterSobelVert\_32f\_C3R
  - fixed\_filters, [1179](#)
- nppiFilterSobelVert\_32f\_C4R
  - fixed\_filters, [1179](#)
- nppiFilterSobelVert\_8s16s\_C1R
  - fixed\_filters, [1180](#)
- nppiFilterSobelVert\_8u16s\_C1R
  - fixed\_filters, [1180](#)
- nppiFilterSobelVert\_8u\_AC4R
  - fixed\_filters, [1180](#)
- nppiFilterSobelVert\_8u\_C1R
  - fixed\_filters, [1181](#)
- nppiFilterSobelVert\_8u\_C3R
  - fixed\_filters, [1181](#)
- nppiFilterSobelVert\_8u\_C4R
  - fixed\_filters, [1181](#)
- nppiFilterSobelVertBorder\_16s\_AC4R
  - image\_1D\_linear\_filter, [1037](#)
- nppiFilterSobelVertBorder\_16s\_C1R
  - image\_1D\_linear\_filter, [1037](#)
- nppiFilterSobelVertBorder\_16s\_C3R
  - image\_1D\_linear\_filter, [1038](#)
- nppiFilterSobelVertBorder\_16s\_C4R
  - image\_1D\_linear\_filter, [1038](#)
- nppiFilterSobelVertBorder\_32f\_AC4R
  - image\_1D\_linear\_filter, [1039](#)
- nppiFilterSobelVertBorder\_32f\_C1R

- image\_1D\_linear\_filter, [1039](#)
- nppiFilterSobelVertBorder\_32f\_C3R
  - image\_1D\_linear\_filter, [1040](#)
- nppiFilterSobelVertBorder\_32f\_C4R
  - image\_1D\_linear\_filter, [1040](#)
- nppiFilterSobelVertBorder\_8s16s\_C1R
  - image\_1D\_linear\_filter, [1040](#)
- nppiFilterSobelVertBorder\_8u16s\_C1R
  - image\_1D\_linear\_filter, [1041](#)
- nppiFilterSobelVertBorder\_8u\_AC4R
  - image\_1D\_linear\_filter, [1041](#)
- nppiFilterSobelVertBorder\_8u\_C1R
  - image\_1D\_linear\_filter, [1042](#)
- nppiFilterSobelVertBorder\_8u\_C3R
  - image\_1D\_linear\_filter, [1042](#)
- nppiFilterSobelVertBorder\_8u\_C4R
  - image\_1D\_linear\_filter, [1043](#)
- nppiFilterSobelVertMask\_32f\_C1R
  - fixed\_filters, [1182](#)
- nppiFilterSobelVertMaskBorder\_32f\_C1R
  - image\_1D\_linear\_filter, [1043](#)
- nppiFilterSobelVertSecond\_32f\_C1R
  - image\_1D\_linear\_filter, [1044](#)
- nppiFilterSobelVertSecond\_8s16s\_C1R
  - image\_1D\_linear\_filter, [1044](#)
- nppiFilterSobelVertSecond\_8u16s\_C1R
  - image\_1D\_linear\_filter, [1044](#)
- nppiFilterSobelVertSecondBorder\_32f\_C1R
  - image\_filtering\_functions, [987](#)
- nppiFilterSobelVertSecondBorder\_8s16s\_C1R
  - image\_filtering\_functions, [987](#)
- nppiFilterSobelVertSecondBorder\_8u16s\_C1R
  - image\_filtering\_functions, [988](#)
- nppiFree
  - image\_memory\_management, [2120](#)
- nppiFullNormLevelGetBufferHostSize\_16u32f\_-AC4R
  - crosscorrfullnormlevel, [1968](#)
- nppiFullNormLevelGetBufferHostSize\_16u32f\_-C1R
  - crosscorrfullnormlevel, [1969](#)
- nppiFullNormLevelGetBufferHostSize\_16u32f\_-C3R
  - crosscorrfullnormlevel, [1969](#)
- nppiFullNormLevelGetBufferHostSize\_16u32f\_-C4R
  - crosscorrfullnormlevel, [1969](#)
- nppiFullNormLevelGetBufferHostSize\_32f\_AC4R
  - crosscorrfullnormlevel, [1970](#)
- nppiFullNormLevelGetBufferHostSize\_32f\_C1R
  - crosscorrfullnormlevel, [1970](#)
- nppiFullNormLevelGetBufferHostSize\_32f\_C3R
  - crosscorrfullnormlevel, [1970](#)
- nppiFullNormLevelGetBufferHostSize\_32f\_C4R
  - crosscorrfullnormlevel, [1970](#)
- nppiFullNormLevelGetBufferHostSize\_8s32f\_-AC4R
  - crosscorrfullnormlevel, [1971](#)
- nppiFullNormLevelGetBufferHostSize\_8s32f\_C1R
  - crosscorrfullnormlevel, [1971](#)
- nppiFullNormLevelGetBufferHostSize\_8s32f\_C3R
  - crosscorrfullnormlevel, [1971](#)
- nppiFullNormLevelGetBufferHostSize\_8s32f\_C4R
  - crosscorrfullnormlevel, [1972](#)
- nppiFullNormLevelGetBufferHostSize\_8u32f\_-AC4R
  - crosscorrfullnormlevel, [1972](#)
- nppiFullNormLevelGetBufferHostSize\_8u32f\_-C1R
  - crosscorrfullnormlevel, [1972](#)
- nppiFullNormLevelGetBufferHostSize\_8u32f\_-C3R
  - crosscorrfullnormlevel, [1972](#)
- nppiFullNormLevelGetBufferHostSize\_8u32f\_-C4R
  - crosscorrfullnormlevel, [1973](#)
- nppiFullNormLevelGetBufferHostSize\_8u\_-AC4RSfs
  - crosscorrfullnormlevel, [1973](#)
- nppiFullNormLevelGetBufferHostSize\_8u\_C1RSfs
  - crosscorrfullnormlevel, [1973](#)
- nppiFullNormLevelGetBufferHostSize\_8u\_C3RSfs
  - crosscorrfullnormlevel, [1974](#)
- nppiFullNormLevelGetBufferHostSize\_8u\_C4RSfs
  - crosscorrfullnormlevel, [1974](#)
- nppiGammaFwd\_8u\_AC4IR
  - image\_color\_gamma\_correction, [607](#)
- nppiGammaFwd\_8u\_AC4R
  - image\_color\_gamma\_correction, [607](#)
- nppiGammaFwd\_8u\_C3IR
  - image\_color\_gamma\_correction, [607](#)
- nppiGammaFwd\_8u\_C3R
  - image\_color\_gamma\_correction, [608](#)
- nppiGammaFwd\_8u\_IP3R
  - image\_color\_gamma\_correction, [608](#)
- nppiGammaFwd\_8u\_P3R
  - image\_color\_gamma\_correction, [608](#)
- nppiGammaInv\_8u\_AC4IR
  - image\_color\_gamma\_correction, [609](#)
- nppiGammaInv\_8u\_AC4R
  - image\_color\_gamma\_correction, [609](#)
- nppiGammaInv\_8u\_C3IR
  - image\_color\_gamma\_correction, [609](#)
- nppiGammaInv\_8u\_C3R
  - image\_color\_gamma\_correction, [610](#)
- nppiGammaInv\_8u\_IP3R
  - image\_color\_gamma\_correction, [610](#)
- nppiGammaInv\_8u\_P3R
  - image\_color\_gamma\_correction, [610](#)

- image\_color\_gamma\_correction, 610
- nppiGetAffineBound
  - image\_affine\_transform, 1276
- nppiGetAffineQuad
  - image\_affine\_transform, 1276
- nppiGetAffineTransform
  - image\_affine\_transform, 1277
- nppiGetPerspectiveBound
  - image\_perspective\_transforms, 1325
- nppiGetPerspectiveQuad
  - image\_perspective\_transforms, 1325
- nppiGetPerspectiveTransform
  - image\_perspective\_transforms, 1326
- nppiGetResizeRect
  - image\_resize\_square\_pixel, 1189
- nppiGetRotateBound
  - image\_rotate, 1242
- nppiGetRotateQuad
  - image\_rotate, 1243
- nppiGraphcut8\_32f8u
  - image\_graphcut, 713
- nppiGraphcut8\_32s8u
  - image\_graphcut, 713
- nppiGraphcut8GetSize
  - image\_graphcut, 714
- nppiGraphcut8InitAlloc
  - image\_graphcut, 715
- nppiGraphcut\_32f8u
  - image\_graphcut, 715
- nppiGraphcut\_32s8u
  - image\_graphcut, 716
- nppiGraphcutFree
  - image\_graphcut, 717
- nppiGraphcutGetSize
  - image\_graphcut, 717
- nppiGraphcutInitAlloc
  - image\_graphcut, 717
- NppiGraphcutState
  - image\_labeling\_and\_segmentation, 711
- NppiHaarBuffer, 2629
  - haarBuffer, 2629
  - haarBufferSize, 2629
- NppiHaarClassifier\_32f, 2630
  - classifiers, 2630
  - classifierSize, 2630
  - classifierStep, 2630
  - counterDevice, 2630
  - numClassifiers, 2630
- nppiHistogramEven\_16s\_AC4R
  - image\_histogrameven, 1857
- nppiHistogramEven\_16s\_C1R
  - image\_histogrameven, 1857
- nppiHistogramEven\_16s\_C3R
  - image\_histogrameven, 1857
- nppiHistogramEven\_16s\_C4R
  - image\_histogrameven, 1858
- nppiHistogramEven\_16u\_AC4R
  - image\_histogrameven, 1858
- nppiHistogramEven\_16u\_C1R
  - image\_histogrameven, 1859
- nppiHistogramEven\_16u\_C3R
  - image\_histogrameven, 1859
- nppiHistogramEven\_16u\_C4R
  - image\_histogrameven, 1860
- nppiHistogramEven\_8u\_AC4R
  - image\_histogrameven, 1860
- nppiHistogramEven\_8u\_C1R
  - image\_histogrameven, 1861
- nppiHistogramEven\_8u\_C3R
  - image\_histogrameven, 1861
- nppiHistogramEven\_8u\_C4R
  - image\_histogrameven, 1862
- nppiHistogramEvenGetBufferSize\_16s\_AC4R
  - image\_histogrameven, 1862
- nppiHistogramEvenGetBufferSize\_16s\_C1R
  - image\_histogrameven, 1862
- nppiHistogramEvenGetBufferSize\_16s\_C3R
  - image\_histogrameven, 1863
- nppiHistogramEvenGetBufferSize\_16s\_C4R
  - image\_histogrameven, 1863
- nppiHistogramEvenGetBufferSize\_16u\_AC4R
  - image\_histogrameven, 1863
- nppiHistogramEvenGetBufferSize\_16u\_C1R
  - image\_histogrameven, 1864
- nppiHistogramEvenGetBufferSize\_16u\_C3R
  - image\_histogrameven, 1864
- nppiHistogramEvenGetBufferSize\_16u\_C4R
  - image\_histogrameven, 1864
- nppiHistogramEvenGetBufferSize\_8u\_AC4R
  - image\_histogrameven, 1865
- nppiHistogramEvenGetBufferSize\_8u\_C1R
  - image\_histogrameven, 1865
- nppiHistogramEvenGetBufferSize\_8u\_C3R
  - image\_histogrameven, 1865
- nppiHistogramEvenGetBufferSize\_8u\_C4R
  - image\_histogrameven, 1866
- nppiHistogramRange\_16s\_AC4R
  - image\_histogramrange, 1870
- nppiHistogramRange\_16s\_C1R
  - image\_histogramrange, 1870
- nppiHistogramRange\_16s\_C3R
  - image\_histogramrange, 1870
- nppiHistogramRange\_16s\_C4R
  - image\_histogramrange, 1871
- nppiHistogramRange\_16u\_AC4R
  - image\_histogramrange, 1871
- nppiHistogramRange\_16u\_C1R
  - image\_histogramrange, 1872



- nppiHistogramRange\_16u\_C3R
  - image\_histogramrange, [1872](#)
- nppiHistogramRange\_16u\_C4R
  - image\_histogramrange, [1873](#)
- nppiHistogramRange\_32f\_AC4R
  - image\_histogramrange, [1873](#)
- nppiHistogramRange\_32f\_C1R
  - image\_histogramrange, [1874](#)
- nppiHistogramRange\_32f\_C3R
  - image\_histogramrange, [1874](#)
- nppiHistogramRange\_32f\_C4R
  - image\_histogramrange, [1874](#)
- nppiHistogramRange\_8u\_AC4R
  - image\_histogramrange, [1875](#)
- nppiHistogramRange\_8u\_C1R
  - image\_histogramrange, [1875](#)
- nppiHistogramRange\_8u\_C3R
  - image\_histogramrange, [1876](#)
- nppiHistogramRange\_8u\_C4R
  - image\_histogramrange, [1876](#)
- nppiHistogramRangeGetBufferSize\_16s\_AC4R
  - image\_histogramrange, [1877](#)
- nppiHistogramRangeGetBufferSize\_16s\_C1R
  - image\_histogramrange, [1877](#)
- nppiHistogramRangeGetBufferSize\_16s\_C3R
  - image\_histogramrange, [1877](#)
- nppiHistogramRangeGetBufferSize\_16s\_C4R
  - image\_histogramrange, [1878](#)
- nppiHistogramRangeGetBufferSize\_16u\_AC4R
  - image\_histogramrange, [1878](#)
- nppiHistogramRangeGetBufferSize\_16u\_C1R
  - image\_histogramrange, [1878](#)
- nppiHistogramRangeGetBufferSize\_16u\_C3R
  - image\_histogramrange, [1879](#)
- nppiHistogramRangeGetBufferSize\_16u\_C4R
  - image\_histogramrange, [1879](#)
- nppiHistogramRangeGetBufferSize\_32f\_AC4R
  - image\_histogramrange, [1879](#)
- nppiHistogramRangeGetBufferSize\_32f\_C1R
  - image\_histogramrange, [1880](#)
- nppiHistogramRangeGetBufferSize\_32f\_C3R
  - image\_histogramrange, [1880](#)
- nppiHistogramRangeGetBufferSize\_32f\_C4R
  - image\_histogramrange, [1880](#)
- nppiHistogramRangeGetBufferSize\_8u\_AC4R
  - image\_histogramrange, [1881](#)
- nppiHistogramRangeGetBufferSize\_8u\_C1R
  - image\_histogramrange, [1881](#)
- nppiHistogramRangeGetBufferSize\_8u\_C3R
  - image\_histogramrange, [1881](#)
- nppiHistogramRangeGetBufferSize\_8u\_C4R
  - image\_histogramrange, [1882](#)
- nppiHLSToBGR\_8u\_AC4P4R
  - image\_color\_model\_conversion, [542](#)
- nppiHLSToBGR\_8u\_AC4R
  - image\_color\_model\_conversion, [543](#)
- nppiHLSToBGR\_8u\_AP4C4R
  - image\_color\_model\_conversion, [543](#)
- nppiHLSToBGR\_8u\_AP4R
  - image\_color\_model\_conversion, [543](#)
- nppiHLSToBGR\_8u\_C3P3R
  - image\_color\_model\_conversion, [544](#)
- nppiHLSToBGR\_8u\_P3C3R
  - image\_color\_model\_conversion, [544](#)
- nppiHLSToBGR\_8u\_P3R
  - image\_color\_model\_conversion, [544](#)
- nppiHLSToRGB\_8u\_AC4R
  - image\_color\_model\_conversion, [545](#)
- nppiHLSToRGB\_8u\_C3R
  - image\_color\_model\_conversion, [545](#)
- nppiHSVToRGB\_8u\_AC4R
  - image\_color\_model\_conversion, [545](#)
- nppiHSVToRGB\_8u\_C3R
  - image\_color\_model\_conversion, [546](#)
- NppiHuffmanTableType
  - typedefs\_npp, [42](#)
- nppiIntegral\_8u32f\_C1R
  - image\_integral, [1846](#)
- nppiIntegral\_8u32s\_C1R
  - image\_integral, [1846](#)
- NppiInterpolationMode
  - typedefs\_npp, [42](#)
- nppiLabToBGR\_8u\_C3R
  - image\_color\_model\_conversion, [546](#)
- nppiLn\_16s\_C1IRSfs
  - image\_ln, [357](#)
- nppiLn\_16s\_C1RSfs
  - image\_ln, [357](#)
- nppiLn\_16s\_C3IRSfs
  - image\_ln, [358](#)
- nppiLn\_16s\_C3RSfs
  - image\_ln, [358](#)
- nppiLn\_16u\_C1IRSfs
  - image\_ln, [358](#)
- nppiLn\_16u\_C1RSfs
  - image\_ln, [359](#)
- nppiLn\_16u\_C3IRSfs
  - image\_ln, [359](#)
- nppiLn\_16u\_C3RSfs
  - image\_ln, [359](#)
- nppiLn\_32f\_C1IR
  - image\_ln, [360](#)
- nppiLn\_32f\_C1R
  - image\_ln, [360](#)
- nppiLn\_32f\_C3IR
  - image\_ln, [360](#)
- nppiLn\_32f\_C3R
  - image\_ln, [361](#)

- nppiLn\_8u\_C1IRSfs
  - image\_lnc, 361
- nppiLn\_8u\_C1RSfs
  - image\_lnc, 361
- nppiLn\_8u\_C3IRSfs
  - image\_lnc, 362
- nppiLn\_8u\_C3RSfs
  - image\_lnc, 362
- nppiLShiftC\_16u\_AC4IR
  - image\_lshiftc, 423
- nppiLShiftC\_16u\_AC4R
  - image\_lshiftc, 423
- nppiLShiftC\_16u\_C1IR
  - image\_lshiftc, 423
- nppiLShiftC\_16u\_C1R
  - image\_lshiftc, 424
- nppiLShiftC\_16u\_C3IR
  - image\_lshiftc, 424
- nppiLShiftC\_16u\_C3R
  - image\_lshiftc, 424
- nppiLShiftC\_16u\_C4IR
  - image\_lshiftc, 425
- nppiLShiftC\_16u\_C4R
  - image\_lshiftc, 425
- nppiLShiftC\_32s\_AC4IR
  - image\_lshiftc, 425
- nppiLShiftC\_32s\_AC4R
  - image\_lshiftc, 426
- nppiLShiftC\_32s\_C1IR
  - image\_lshiftc, 426
- nppiLShiftC\_32s\_C1R
  - image\_lshiftc, 426
- nppiLShiftC\_32s\_C3IR
  - image\_lshiftc, 427
- nppiLShiftC\_32s\_C3R
  - image\_lshiftc, 427
- nppiLShiftC\_32s\_C4IR
  - image\_lshiftc, 427
- nppiLShiftC\_32s\_C4R
  - image\_lshiftc, 428
- nppiLShiftC\_8u\_AC4IR
  - image\_lshiftc, 428
- nppiLShiftC\_8u\_AC4R
  - image\_lshiftc, 428
- nppiLShiftC\_8u\_C1IR
  - image\_lshiftc, 429
- nppiLShiftC\_8u\_C1R
  - image\_lshiftc, 429
- nppiLShiftC\_8u\_C3IR
  - image\_lshiftc, 429
- nppiLShiftC\_8u\_C3R
  - image\_lshiftc, 430
- nppiLShiftC\_8u\_C4IR
  - image\_lshiftc, 430
- nppiLShiftC\_8u\_C4R
  - image\_lshiftc, 430
- nppiLUT\_16s\_AC4IR
  - image\_color\_processing, 641
- nppiLUT\_16s\_AC4R
  - image\_color\_processing, 642
- nppiLUT\_16s\_C1IR
  - image\_color\_processing, 642
- nppiLUT\_16s\_C1R
  - image\_color\_processing, 643
- nppiLUT\_16s\_C3IR
  - image\_color\_processing, 643
- nppiLUT\_16s\_C3R
  - image\_color\_processing, 644
- nppiLUT\_16s\_C4IR
  - image\_color\_processing, 644
- nppiLUT\_16s\_C4R
  - image\_color\_processing, 645
- nppiLUT\_16u\_AC4IR
  - image\_color\_processing, 645
- nppiLUT\_16u\_AC4R
  - image\_color\_processing, 646
- nppiLUT\_16u\_C1IR
  - image\_color\_processing, 646
- nppiLUT\_16u\_C1R
  - image\_color\_processing, 647
- nppiLUT\_16u\_C3IR
  - image\_color\_processing, 647
- nppiLUT\_16u\_C3R
  - image\_color\_processing, 648
- nppiLUT\_16u\_C4IR
  - image\_color\_processing, 648
- nppiLUT\_16u\_C4R
  - image\_color\_processing, 649
- nppiLUT\_32f\_AC4IR
  - image\_color\_processing, 649
- nppiLUT\_32f\_AC4R
  - image\_color\_processing, 650
- nppiLUT\_32f\_C1IR
  - image\_color\_processing, 650
- nppiLUT\_32f\_C1R
  - image\_color\_processing, 651
- nppiLUT\_32f\_C3IR
  - image\_color\_processing, 651
- nppiLUT\_32f\_C3R
  - image\_color\_processing, 652
- nppiLUT\_32f\_C4IR
  - image\_color\_processing, 652
- nppiLUT\_32f\_C4R
  - image\_color\_processing, 653
- nppiLUT\_8u\_AC4IR
  - image\_color\_processing, 653
- nppiLUT\_8u\_AC4R
  - image\_color\_processing, 654

- nppiLUT\_8u\_C1IR
  - image\_color\_processing, [654](#)
- nppiLUT\_8u\_C1R
  - image\_color\_processing, [655](#)
- nppiLUT\_8u\_C3IR
  - image\_color\_processing, [655](#)
- nppiLUT\_8u\_C3R
  - image\_color\_processing, [656](#)
- nppiLUT\_8u\_C4IR
  - image\_color\_processing, [656](#)
- nppiLUT\_8u\_C4R
  - image\_color\_processing, [657](#)
- nppiLUT\_Cubic\_16s\_AC4IR
  - image\_color\_processing, [657](#)
- nppiLUT\_Cubic\_16s\_AC4R
  - image\_color\_processing, [658](#)
- nppiLUT\_Cubic\_16s\_C1IR
  - image\_color\_processing, [658](#)
- nppiLUT\_Cubic\_16s\_C1R
  - image\_color\_processing, [659](#)
- nppiLUT\_Cubic\_16s\_C3IR
  - image\_color\_processing, [659](#)
- nppiLUT\_Cubic\_16s\_C3R
  - image\_color\_processing, [660](#)
- nppiLUT\_Cubic\_16s\_C4IR
  - image\_color\_processing, [660](#)
- nppiLUT\_Cubic\_16s\_C4R
  - image\_color\_processing, [661](#)
- nppiLUT\_Cubic\_16u\_AC4IR
  - image\_color\_processing, [661](#)
- nppiLUT\_Cubic\_16u\_AC4R
  - image\_color\_processing, [662](#)
- nppiLUT\_Cubic\_16u\_C1IR
  - image\_color\_processing, [662](#)
- nppiLUT\_Cubic\_16u\_C1R
  - image\_color\_processing, [663](#)
- nppiLUT\_Cubic\_16u\_C3IR
  - image\_color\_processing, [663](#)
- nppiLUT\_Cubic\_16u\_C3R
  - image\_color\_processing, [664](#)
- nppiLUT\_Cubic\_16u\_C4IR
  - image\_color\_processing, [664](#)
- nppiLUT\_Cubic\_16u\_C4R
  - image\_color\_processing, [665](#)
- nppiLUT\_Cubic\_32f\_AC4IR
  - image\_color\_processing, [665](#)
- nppiLUT\_Cubic\_32f\_AC4R
  - image\_color\_processing, [666](#)
- nppiLUT\_Cubic\_32f\_C1IR
  - image\_color\_processing, [666](#)
- nppiLUT\_Cubic\_32f\_C1R
  - image\_color\_processing, [667](#)
- nppiLUT\_Cubic\_32f\_C3IR
  - image\_color\_processing, [667](#)
- nppiLUT\_Cubic\_32f\_C3R
  - image\_color\_processing, [668](#)
- nppiLUT\_Cubic\_32f\_C4IR
  - image\_color\_processing, [668](#)
- nppiLUT\_Cubic\_32f\_C4R
  - image\_color\_processing, [669](#)
- nppiLUT\_Cubic\_8u\_AC4IR
  - image\_color\_processing, [669](#)
- nppiLUT\_Cubic\_8u\_AC4R
  - image\_color\_processing, [670](#)
- nppiLUT\_Cubic\_8u\_C1IR
  - image\_color\_processing, [670](#)
- nppiLUT\_Cubic\_8u\_C1R
  - image\_color\_processing, [671](#)
- nppiLUT\_Cubic\_8u\_C3IR
  - image\_color\_processing, [671](#)
- nppiLUT\_Cubic\_8u\_C3R
  - image\_color\_processing, [672](#)
- nppiLUT\_Cubic\_8u\_C4IR
  - image\_color\_processing, [672](#)
- nppiLUT\_Cubic\_8u\_C4R
  - image\_color\_processing, [673](#)
- nppiLUT\_Linear\_16s\_AC4IR
  - image\_color\_processing, [673](#)
- nppiLUT\_Linear\_16s\_AC4R
  - image\_color\_processing, [674](#)
- nppiLUT\_Linear\_16s\_C1IR
  - image\_color\_processing, [675](#)
- nppiLUT\_Linear\_16s\_C1R
  - image\_color\_processing, [675](#)
- nppiLUT\_Linear\_16s\_C3IR
  - image\_color\_processing, [675](#)
- nppiLUT\_Linear\_16s\_C3R
  - image\_color\_processing, [676](#)
- nppiLUT\_Linear\_16s\_C4IR
  - image\_color\_processing, [676](#)
- nppiLUT\_Linear\_16s\_C4R
  - image\_color\_processing, [677](#)
- nppiLUT\_Linear\_16u\_AC4IR
  - image\_color\_processing, [677](#)
- nppiLUT\_Linear\_16u\_AC4R
  - image\_color\_processing, [678](#)
- nppiLUT\_Linear\_16u\_C1IR
  - image\_color\_processing, [679](#)
- nppiLUT\_Linear\_16u\_C1R
  - image\_color\_processing, [679](#)
- nppiLUT\_Linear\_16u\_C3IR
  - image\_color\_processing, [679](#)
- nppiLUT\_Linear\_16u\_C3R
  - image\_color\_processing, [680](#)
- nppiLUT\_Linear\_16u\_C4IR
  - image\_color\_processing, [680](#)
- nppiLUT\_Linear\_16u\_C4R
  - image\_color\_processing, [681](#)



- nppiLUT\_Linear\_32f\_AC4IR
  - image\_color\_processing, [681](#)
- nppiLUT\_Linear\_32f\_AC4R
  - image\_color\_processing, [682](#)
- nppiLUT\_Linear\_32f\_C1IR
  - image\_color\_processing, [682](#)
- nppiLUT\_Linear\_32f\_C1R
  - image\_color\_processing, [683](#)
- nppiLUT\_Linear\_32f\_C3IR
  - image\_color\_processing, [683](#)
- nppiLUT\_Linear\_32f\_C3R
  - image\_color\_processing, [684](#)
- nppiLUT\_Linear\_32f\_C4IR
  - image\_color\_processing, [684](#)
- nppiLUT\_Linear\_32f\_C4R
  - image\_color\_processing, [685](#)
- nppiLUT\_Linear\_8u\_AC4IR
  - image\_color\_processing, [685](#)
- nppiLUT\_Linear\_8u\_AC4R
  - image\_color\_processing, [686](#)
- nppiLUT\_Linear\_8u\_C1IR
  - image\_color\_processing, [687](#)
- nppiLUT\_Linear\_8u\_C1R
  - image\_color\_processing, [687](#)
- nppiLUT\_Linear\_8u\_C3IR
  - image\_color\_processing, [688](#)
- nppiLUT\_Linear\_8u\_C3R
  - image\_color\_processing, [688](#)
- nppiLUT\_Linear\_8u\_C4IR
  - image\_color\_processing, [689](#)
- nppiLUT\_Linear\_8u\_C4R
  - image\_color\_processing, [689](#)
- nppiLUT\_Trilinear\_8u\_AC4IR
  - image\_color\_processing, [690](#)
- nppiLUT\_Trilinear\_8u\_AC4R
  - image\_color\_processing, [691](#)
- nppiLUT\_Trilinear\_8u\_C4R
  - image\_color\_processing, [691](#)
- nppiLUTPalette\_16u24u\_C1R
  - image\_color\_processing, [692](#)
- nppiLUTPalette\_16u32u\_C1R
  - image\_color\_processing, [692](#)
- nppiLUTPalette\_16u8u\_C1R
  - image\_color\_processing, [693](#)
- nppiLUTPalette\_16u\_AC4R
  - image\_color\_processing, [693](#)
- nppiLUTPalette\_16u\_C1R
  - image\_color\_processing, [694](#)
- nppiLUTPalette\_16u\_C3R
  - image\_color\_processing, [694](#)
- nppiLUTPalette\_16u\_C4R
  - image\_color\_processing, [695](#)
- nppiLUTPalette\_8u24u\_C1R
  - image\_color\_processing, [695](#)
- nppiLUTPalette\_8u32u\_C1R
  - image\_color\_processing, [696](#)
- nppiLUTPalette\_8u\_AC4R
  - image\_color\_processing, [696](#)
- nppiLUTPalette\_8u\_C1R
  - image\_color\_processing, [697](#)
- nppiLUTPalette\_8u\_C3R
  - image\_color\_processing, [697](#)
- nppiLUTPalette\_8u\_C4R
  - image\_color\_processing, [698](#)
- nppiLUTPaletteSwap\_16u\_C3A0C4R
  - image\_color\_processing, [698](#)
- nppiLUTPaletteSwap\_8u\_C3A0C4R
  - image\_color\_processing, [699](#)
- nppiLUVToRGB\_8u\_AC4R
  - image\_color\_model\_conversion, [546](#)
- nppiLUVToRGB\_8u\_C3R
  - image\_color\_model\_conversion, [547](#)
- nppiMagnitude\_32fc32f\_C1R
  - image\_fourier\_transforms, [1364](#)
- nppiMagnitudeSqr\_32fc32f\_C1R
  - image\_fourier\_transforms, [1364](#)
- nppiMalloc\_16s\_C1
  - image\_memory\_management, [2120](#)
- nppiMalloc\_16s\_C2
  - image\_memory\_management, [2120](#)
- nppiMalloc\_16s\_C4
  - image\_memory\_management, [2121](#)
- nppiMalloc\_16sc\_C1
  - image\_memory\_management, [2121](#)
- nppiMalloc\_16sc\_C2
  - image\_memory\_management, [2121](#)
- nppiMalloc\_16sc\_C3
  - image\_memory\_management, [2122](#)
- nppiMalloc\_16sc\_C4
  - image\_memory\_management, [2122](#)
- nppiMalloc\_16u\_C1
  - image\_memory\_management, [2122](#)
- nppiMalloc\_16u\_C2
  - image\_memory\_management, [2122](#)
- nppiMalloc\_16u\_C3
  - image\_memory\_management, [2123](#)
- nppiMalloc\_16u\_C4
  - image\_memory\_management, [2123](#)
- nppiMalloc\_32f\_C1
  - image\_memory\_management, [2123](#)
- nppiMalloc\_32f\_C2
  - image\_memory\_management, [2124](#)
- nppiMalloc\_32f\_C3
  - image\_memory\_management, [2124](#)
- nppiMalloc\_32f\_C4
  - image\_memory\_management, [2124](#)
- nppiMalloc\_32fc\_C1
  - image\_memory\_management, [2124](#)

- nppiMalloc\_32fc\_C2
  - image\_memory\_management, [2125](#)
- nppiMalloc\_32fc\_C3
  - image\_memory\_management, [2125](#)
- nppiMalloc\_32fc\_C4
  - image\_memory\_management, [2125](#)
- nppiMalloc\_32s\_C1
  - image\_memory\_management, [2126](#)
- nppiMalloc\_32s\_C3
  - image\_memory\_management, [2126](#)
- nppiMalloc\_32s\_C4
  - image\_memory\_management, [2126](#)
- nppiMalloc\_32sc\_C1
  - image\_memory\_management, [2126](#)
- nppiMalloc\_32sc\_C2
  - image\_memory\_management, [2127](#)
- nppiMalloc\_32sc\_C3
  - image\_memory\_management, [2127](#)
- nppiMalloc\_32sc\_C4
  - image\_memory\_management, [2127](#)
- nppiMalloc\_8u\_C1
  - image\_memory\_management, [2128](#)
- nppiMalloc\_8u\_C2
  - image\_memory\_management, [2128](#)
- nppiMalloc\_8u\_C3
  - image\_memory\_management, [2128](#)
- nppiMalloc\_8u\_C4
  - image\_memory\_management, [2128](#)
- NppiMaskSize
  - typedefs\_npp, [43](#)
- nppiMax\_16s\_AC4R
  - image\_max, [1504](#)
- nppiMax\_16s\_C1R
  - image\_max, [1504](#)
- nppiMax\_16s\_C3R
  - image\_max, [1505](#)
- nppiMax\_16s\_C4R
  - image\_max, [1505](#)
- nppiMax\_16u\_AC4R
  - image\_max, [1505](#)
- nppiMax\_16u\_C1R
  - image\_max, [1506](#)
- nppiMax\_16u\_C3R
  - image\_max, [1506](#)
- nppiMax\_16u\_C4R
  - image\_max, [1507](#)
- nppiMax\_32f\_AC4R
  - image\_max, [1507](#)
- nppiMax\_32f\_C1R
  - image\_max, [1507](#)
- nppiMax\_32f\_C3R
  - image\_max, [1508](#)
- nppiMax\_32f\_C4R
  - image\_max, [1508](#)
- nppiMax\_8u\_AC4R
  - image\_max, [1508](#)
- nppiMax\_8u\_C1R
  - image\_max, [1509](#)
- nppiMax\_8u\_C3R
  - image\_max, [1509](#)
- nppiMax\_8u\_C4R
  - image\_max, [1510](#)
- nppiMaxEvery\_16s\_AC4IR
  - image\_maxevery, [1833](#)
- nppiMaxEvery\_16s\_C1IR
  - image\_maxevery, [1833](#)
- nppiMaxEvery\_16s\_C3IR
  - image\_maxevery, [1834](#)
- nppiMaxEvery\_16s\_C4IR
  - image\_maxevery, [1834](#)
- nppiMaxEvery\_16u\_AC4IR
  - image\_maxevery, [1834](#)
- nppiMaxEvery\_16u\_C1IR
  - image\_maxevery, [1835](#)
- nppiMaxEvery\_16u\_C3IR
  - image\_maxevery, [1835](#)
- nppiMaxEvery\_16u\_C4IR
  - image\_maxevery, [1835](#)
- nppiMaxEvery\_32f\_AC4IR
  - image\_maxevery, [1836](#)
- nppiMaxEvery\_32f\_C1IR
  - image\_maxevery, [1836](#)
- nppiMaxEvery\_32f\_C3IR
  - image\_maxevery, [1836](#)
- nppiMaxEvery\_32f\_C4IR
  - image\_maxevery, [1837](#)
- nppiMaxEvery\_8u\_AC4IR
  - image\_maxevery, [1837](#)
- nppiMaxEvery\_8u\_C1IR
  - image\_maxevery, [1837](#)
- nppiMaxEvery\_8u\_C3IR
  - image\_maxevery, [1838](#)
- nppiMaxEvery\_8u\_C4IR
  - image\_maxevery, [1838](#)
- nppiMaxGetBufferHostSize\_16s\_AC4R
  - image\_max, [1510](#)
- nppiMaxGetBufferHostSize\_16s\_C1R
  - image\_max, [1510](#)
- nppiMaxGetBufferHostSize\_16s\_C3R
  - image\_max, [1510](#)
- nppiMaxGetBufferHostSize\_16s\_C4R
  - image\_max, [1511](#)
- nppiMaxGetBufferHostSize\_16u\_AC4R
  - image\_max, [1511](#)
- nppiMaxGetBufferHostSize\_16u\_C1R
  - image\_max, [1511](#)
- nppiMaxGetBufferHostSize\_16u\_C3R
  - image\_max, [1512](#)

- nppiMaxGetBufferHostSize\_16u\_C4R
  - image\_max, [1512](#)
- nppiMaxGetBufferHostSize\_32f\_AC4R
  - image\_max, [1512](#)
- nppiMaxGetBufferHostSize\_32f\_C1R
  - image\_max, [1512](#)
- nppiMaxGetBufferHostSize\_32f\_C3R
  - image\_max, [1513](#)
- nppiMaxGetBufferHostSize\_32f\_C4R
  - image\_max, [1513](#)
- nppiMaxGetBufferHostSize\_8u\_AC4R
  - image\_max, [1513](#)
- nppiMaxGetBufferHostSize\_8u\_C1R
  - image\_max, [1514](#)
- nppiMaxGetBufferHostSize\_8u\_C3R
  - image\_max, [1514](#)
- nppiMaxGetBufferHostSize\_8u\_C4R
  - image\_max, [1514](#)
- nppiMaximumError\_16s\_C1R
  - image\_maximum\_error, [2027](#)
- nppiMaximumError\_16s\_C2R
  - image\_maximum\_error, [2028](#)
- nppiMaximumError\_16s\_C3R
  - image\_maximum\_error, [2028](#)
- nppiMaximumError\_16s\_C4R
  - image\_maximum\_error, [2028](#)
- nppiMaximumError\_16sc\_C1R
  - image\_maximum\_error, [2029](#)
- nppiMaximumError\_16sc\_C2R
  - image\_maximum\_error, [2029](#)
- nppiMaximumError\_16sc\_C3R
  - image\_maximum\_error, [2030](#)
- nppiMaximumError\_16sc\_C4R
  - image\_maximum\_error, [2030](#)
- nppiMaximumError\_16u\_C1R
  - image\_maximum\_error, [2031](#)
- nppiMaximumError\_16u\_C2R
  - image\_maximum\_error, [2031](#)
- nppiMaximumError\_16u\_C3R
  - image\_maximum\_error, [2031](#)
- nppiMaximumError\_16u\_C4R
  - image\_maximum\_error, [2032](#)
- nppiMaximumError\_32f\_C1R
  - image\_maximum\_error, [2032](#)
- nppiMaximumError\_32f\_C2R
  - image\_maximum\_error, [2033](#)
- nppiMaximumError\_32f\_C3R
  - image\_maximum\_error, [2033](#)
- nppiMaximumError\_32f\_C4R
  - image\_maximum\_error, [2034](#)
- nppiMaximumError\_32fc\_C1R
  - image\_maximum\_error, [2034](#)
- nppiMaximumError\_32fc\_C2R
  - image\_maximum\_error, [2035](#)
- nppiMaximumError\_32fc\_C3R
  - image\_maximum\_error, [2035](#)
- nppiMaximumError\_32fc\_C4R
  - image\_maximum\_error, [2035](#)
- nppiMaximumError\_32s\_C1R
  - image\_maximum\_error, [2036](#)
- nppiMaximumError\_32s\_C2R
  - image\_maximum\_error, [2036](#)
- nppiMaximumError\_32s\_C3R
  - image\_maximum\_error, [2037](#)
- nppiMaximumError\_32s\_C4R
  - image\_maximum\_error, [2037](#)
- nppiMaximumError\_32sc\_C1R
  - image\_maximum\_error, [2038](#)
- nppiMaximumError\_32sc\_C2R
  - image\_maximum\_error, [2038](#)
- nppiMaximumError\_32sc\_C3R
  - image\_maximum\_error, [2038](#)
- nppiMaximumError\_32sc\_C4R
  - image\_maximum\_error, [2039](#)
- nppiMaximumError\_32u\_C1R
  - image\_maximum\_error, [2039](#)
- nppiMaximumError\_32u\_C2R
  - image\_maximum\_error, [2040](#)
- nppiMaximumError\_32u\_C3R
  - image\_maximum\_error, [2040](#)
- nppiMaximumError\_32u\_C4R
  - image\_maximum\_error, [2041](#)
- nppiMaximumError\_64f\_C1R
  - image\_maximum\_error, [2041](#)
- nppiMaximumError\_64f\_C2R
  - image\_maximum\_error, [2041](#)
- nppiMaximumError\_64f\_C3R
  - image\_maximum\_error, [2042](#)
- nppiMaximumError\_64f\_C4R
  - image\_maximum\_error, [2042](#)
- nppiMaximumError\_8s\_C1R
  - image\_maximum\_error, [2043](#)
- nppiMaximumError\_8s\_C2R
  - image\_maximum\_error, [2043](#)
- nppiMaximumError\_8s\_C3R
  - image\_maximum\_error, [2044](#)
- nppiMaximumError\_8s\_C4R
  - image\_maximum\_error, [2044](#)
- nppiMaximumError\_8u\_C1R
  - image\_maximum\_error, [2044](#)
- nppiMaximumError\_8u\_C2R
  - image\_maximum\_error, [2045](#)
- nppiMaximumError\_8u\_C3R
  - image\_maximum\_error, [2045](#)
- nppiMaximumError\_8u\_C4R
  - image\_maximum\_error, [2046](#)
- nppiMaximumErrorGetBufferHostSize\_16s\_C1R
  - image\_statistics\_functions, [1434](#)



- nppiMaximumRelativeError\_16u\_C4R
  - image\_maximum\_relative\_error, [2078](#)
- nppiMaximumRelativeError\_32f\_C1R
  - image\_maximum\_relative\_error, [2079](#)
- nppiMaximumRelativeError\_32f\_C2R
  - image\_maximum\_relative\_error, [2079](#)
- nppiMaximumRelativeError\_32f\_C3R
  - image\_maximum\_relative\_error, [2080](#)
- nppiMaximumRelativeError\_32f\_C4R
  - image\_maximum\_relative\_error, [2080](#)
- nppiMaximumRelativeError\_32fc\_C1R
  - image\_maximum\_relative\_error, [2081](#)
- nppiMaximumRelativeError\_32fc\_C2R
  - image\_maximum\_relative\_error, [2081](#)
- nppiMaximumRelativeError\_32fc\_C3R
  - image\_maximum\_relative\_error, [2081](#)
- nppiMaximumRelativeError\_32fc\_C4R
  - image\_maximum\_relative\_error, [2082](#)
- nppiMaximumRelativeError\_32s\_C1R
  - image\_maximum\_relative\_error, [2082](#)
- nppiMaximumRelativeError\_32s\_C2R
  - image\_maximum\_relative\_error, [2083](#)
- nppiMaximumRelativeError\_32s\_C3R
  - image\_maximum\_relative\_error, [2083](#)
- nppiMaximumRelativeError\_32s\_C4R
  - image\_maximum\_relative\_error, [2084](#)
- nppiMaximumRelativeError\_32sc\_C1R
  - image\_maximum\_relative\_error, [2084](#)
- nppiMaximumRelativeError\_32sc\_C2R
  - image\_maximum\_relative\_error, [2085](#)
- nppiMaximumRelativeError\_32sc\_C3R
  - image\_maximum\_relative\_error, [2085](#)
- nppiMaximumRelativeError\_32sc\_C4R
  - image\_maximum\_relative\_error, [2086](#)
- nppiMaximumRelativeError\_32u\_C1R
  - image\_maximum\_relative\_error, [2086](#)
- nppiMaximumRelativeError\_32u\_C2R
  - image\_maximum\_relative\_error, [2086](#)
- nppiMaximumRelativeError\_32u\_C3R
  - image\_maximum\_relative\_error, [2087](#)
- nppiMaximumRelativeError\_32u\_C4R
  - image\_maximum\_relative\_error, [2087](#)
- nppiMaximumRelativeError\_64f\_C1R
  - image\_maximum\_relative\_error, [2088](#)
- nppiMaximumRelativeError\_64f\_C2R
  - image\_maximum\_relative\_error, [2088](#)
- nppiMaximumRelativeError\_64f\_C3R
  - image\_maximum\_relative\_error, [2089](#)
- nppiMaximumRelativeError\_64f\_C4R
  - image\_maximum\_relative\_error, [2089](#)
- nppiMaximumRelativeError\_8s\_C1R
  - image\_maximum\_relative\_error, [2090](#)
- nppiMaximumRelativeError\_8s\_C2R
  - image\_maximum\_relative\_error, [2090](#)
- nppiMaximumRelativeError\_8s\_C3R
  - image\_maximum\_relative\_error, [2091](#)
- nppiMaximumRelativeError\_8s\_C4R
  - image\_maximum\_relative\_error, [2091](#)
- nppiMaximumRelativeError\_8u\_C1R
  - image\_maximum\_relative\_error, [2091](#)
- nppiMaximumRelativeError\_8u\_C2R
  - image\_maximum\_relative\_error, [2092](#)
- nppiMaximumRelativeError\_8u\_C3R
  - image\_maximum\_relative\_error, [2092](#)
- nppiMaximumRelativeError\_8u\_C4R
  - image\_maximum\_relative\_error, [2093](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16s\_C1R
  - image\_statistics\_functions, [1446](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16s\_C2R
  - image\_statistics\_functions, [1447](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16s\_C3R
  - image\_statistics\_functions, [1447](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16s\_C4R
  - image\_statistics\_functions, [1447](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16sc\_C1R
  - image\_statistics\_functions, [1447](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16sc\_C2R
  - image\_statistics\_functions, [1448](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16sc\_C3R
  - image\_statistics\_functions, [1448](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16sc\_C4R
  - image\_statistics\_functions, [1448](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16u\_C1R
  - image\_statistics\_functions, [1449](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16u\_C2R
  - image\_statistics\_functions, [1449](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16u\_C3R
  - image\_statistics\_functions, [1449](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-16u\_C4R
  - image\_statistics\_functions, [1449](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-32f\_C1R
  - image\_statistics\_functions, [1450](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-32f\_C2R
  - image\_statistics\_functions, [1450](#)



- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32f\_C3R  
image\_statistics\_functions, [1450](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32f\_C4R  
image\_statistics\_functions, [1451](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32fc\_C1R  
image\_statistics\_functions, [1451](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32fc\_C2R  
image\_statistics\_functions, [1451](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32fc\_C3R  
image\_statistics\_functions, [1451](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32fc\_C4R  
image\_statistics\_functions, [1452](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32s\_C1R  
image\_statistics\_functions, [1452](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32s\_C2R  
image\_statistics\_functions, [1452](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32s\_C3R  
image\_statistics\_functions, [1453](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32s\_C4R  
image\_statistics\_functions, [1453](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32sc\_C1R  
image\_statistics\_functions, [1453](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32sc\_C2R  
image\_statistics\_functions, [1453](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32sc\_C3R  
image\_statistics\_functions, [1454](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32sc\_C4R  
image\_statistics\_functions, [1454](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32u\_C1R  
image\_statistics\_functions, [1454](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32u\_C2R  
image\_statistics\_functions, [1455](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32u\_C3R  
image\_statistics\_functions, [1455](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
32u\_C4R  
image\_statistics\_functions, [1455](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
64f\_C1R  
image\_statistics\_functions, [1455](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
64f\_C2R  
image\_statistics\_functions, [1456](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
64f\_C3R  
image\_statistics\_functions, [1456](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
64f\_C4R  
image\_statistics\_functions, [1456](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8s\_C1R  
image\_statistics\_functions, [1457](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8s\_C2R  
image\_statistics\_functions, [1457](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8s\_C3R  
image\_statistics\_functions, [1457](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8s\_C4R  
image\_statistics\_functions, [1457](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8u\_C1R  
image\_statistics\_functions, [1458](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8u\_C2R  
image\_statistics\_functions, [1458](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8u\_C3R  
image\_statistics\_functions, [1458](#)
- nppiMaximumRelativeErrorGetBufferHostSize\_-  
8u\_C4R  
image\_statistics\_functions, [1459](#)
- nppiMaxIdx\_16s\_AC4R  
image\_max\_index, [1517](#)
- nppiMaxIdx\_16s\_C1R  
image\_max\_index, [1518](#)
- nppiMaxIdx\_16s\_C3R  
image\_max\_index, [1518](#)
- nppiMaxIdx\_16s\_C4R  
image\_max\_index, [1518](#)
- nppiMaxIdx\_16u\_AC4R  
image\_max\_index, [1519](#)
- nppiMaxIdx\_16u\_C1R  
image\_max\_index, [1519](#)
- nppiMaxIdx\_16u\_C3R  
image\_max\_index, [1520](#)
- nppiMaxIdx\_16u\_C4R  
image\_max\_index, [1520](#)
- nppiMaxIdx\_32f\_AC4R  
image\_max\_index, [1520](#)

- nppiMaxIndx\_32f\_C1R
  - image\_max\_index, [1521](#)
- nppiMaxIndx\_32f\_C3R
  - image\_max\_index, [1521](#)
- nppiMaxIndx\_32f\_C4R
  - image\_max\_index, [1522](#)
- nppiMaxIndx\_8u\_AC4R
  - image\_max\_index, [1522](#)
- nppiMaxIndx\_8u\_C1R
  - image\_max\_index, [1522](#)
- nppiMaxIndx\_8u\_C3R
  - image\_max\_index, [1523](#)
- nppiMaxIndx\_8u\_C4R
  - image\_max\_index, [1523](#)
- nppiMaxIndxGetBufferHostSize\_16s\_AC4R
  - image\_max\_index, [1524](#)
- nppiMaxIndxGetBufferHostSize\_16s\_C1R
  - image\_max\_index, [1524](#)
- nppiMaxIndxGetBufferHostSize\_16s\_C3R
  - image\_max\_index, [1524](#)
- nppiMaxIndxGetBufferHostSize\_16s\_C4R
  - image\_max\_index, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_AC4R
  - image\_max\_index, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_C1R
  - image\_max\_index, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_C3R
  - image\_max\_index, [1525](#)
- nppiMaxIndxGetBufferHostSize\_16u\_C4R
  - image\_max\_index, [1526](#)
- nppiMaxIndxGetBufferHostSize\_32f\_AC4R
  - image\_max\_index, [1526](#)
- nppiMaxIndxGetBufferHostSize\_32f\_C1R
  - image\_max\_index, [1526](#)
- nppiMaxIndxGetBufferHostSize\_32f\_C3R
  - image\_max\_index, [1527](#)
- nppiMaxIndxGetBufferHostSize\_32f\_C4R
  - image\_max\_index, [1527](#)
- nppiMaxIndxGetBufferHostSize\_8u\_AC4R
  - image\_max\_index, [1527](#)
- nppiMaxIndxGetBufferHostSize\_8u\_C1R
  - image\_max\_index, [1527](#)
- nppiMaxIndxGetBufferHostSize\_8u\_C3R
  - image\_max\_index, [1528](#)
- nppiMaxIndxGetBufferHostSize\_8u\_C4R
  - image\_max\_index, [1528](#)
- nppiMean\_16s\_AC4R
  - image\_mean, [1564](#)
- nppiMean\_16s\_C1R
  - image\_mean, [1564](#)
- nppiMean\_16s\_C3R
  - image\_mean, [1564](#)
- nppiMean\_16s\_C4R
  - image\_mean, [1565](#)
- nppiMean\_16u\_AC4R
  - image\_mean, [1565](#)
- nppiMean\_16u\_C1MR
  - image\_mean, [1565](#)
- nppiMean\_16u\_C1R
  - image\_mean, [1566](#)
- nppiMean\_16u\_C3CMR
  - image\_mean, [1566](#)
- nppiMean\_16u\_C3R
  - image\_mean, [1566](#)
- nppiMean\_16u\_C4R
  - image\_mean, [1567](#)
- nppiMean\_32f\_AC4R
  - image\_mean, [1567](#)
- nppiMean\_32f\_C1MR
  - image\_mean, [1568](#)
- nppiMean\_32f\_C1R
  - image\_mean, [1568](#)
- nppiMean\_32f\_C3CMR
  - image\_mean, [1568](#)
- nppiMean\_32f\_C3R
  - image\_mean, [1569](#)
- nppiMean\_32f\_C4R
  - image\_mean, [1569](#)
- nppiMean\_8s\_C1MR
  - image\_mean, [1570](#)
- nppiMean\_8s\_C3CMR
  - image\_mean, [1570](#)
- nppiMean\_8u\_AC4R
  - image\_mean, [1571](#)
- nppiMean\_8u\_C1MR
  - image\_mean, [1571](#)
- nppiMean\_8u\_C1R
  - image\_mean, [1571](#)
- nppiMean\_8u\_C3CMR
  - image\_mean, [1572](#)
- nppiMean\_8u\_C3R
  - image\_mean, [1572](#)
- nppiMean\_8u\_C4R
  - image\_mean, [1573](#)
- nppiMean\_StdDev\_16u\_C1MR
  - image\_mean\_stddev, [1584](#)
- nppiMean\_StdDev\_16u\_C1R
  - image\_mean\_stddev, [1584](#)
- nppiMean\_StdDev\_16u\_C3CMR
  - image\_mean\_stddev, [1585](#)
- nppiMean\_StdDev\_16u\_C3CR
  - image\_mean\_stddev, [1585](#)
- nppiMean\_StdDev\_32f\_C1MR
  - image\_mean\_stddev, [1586](#)
- nppiMean\_StdDev\_32f\_C1R
  - image\_mean\_stddev, [1586](#)
- nppiMean\_StdDev\_32f\_C3CMR
  - image\_mean\_stddev, [1587](#)

- nppiMean\_StdDev\_32f\_C3CR  
image\_mean\_stddev, [1587](#)
- nppiMean\_StdDev\_8s\_C1MR  
image\_mean\_stddev, [1588](#)
- nppiMean\_StdDev\_8s\_C1R  
image\_mean\_stddev, [1588](#)
- nppiMean\_StdDev\_8s\_C3CMR  
image\_mean\_stddev, [1589](#)
- nppiMean\_StdDev\_8s\_C3CR  
image\_mean\_stddev, [1589](#)
- nppiMean\_StdDev\_8u\_C1MR  
image\_mean\_stddev, [1590](#)
- nppiMean\_StdDev\_8u\_C1R  
image\_mean\_stddev, [1590](#)
- nppiMean\_StdDev\_8u\_C3CMR  
image\_mean\_stddev, [1591](#)
- nppiMean\_StdDev\_8u\_C3CR  
image\_mean\_stddev, [1591](#)
- nppiMeanGetBufferHostSize\_16s\_AC4R  
image\_mean, [1573](#)
- nppiMeanGetBufferHostSize\_16s\_C1R  
image\_mean, [1573](#)
- nppiMeanGetBufferHostSize\_16s\_C3R  
image\_mean, [1574](#)
- nppiMeanGetBufferHostSize\_16s\_C4R  
image\_mean, [1574](#)
- nppiMeanGetBufferHostSize\_16u\_AC4R  
image\_mean, [1574](#)
- nppiMeanGetBufferHostSize\_16u\_C1MR  
image\_mean, [1574](#)
- nppiMeanGetBufferHostSize\_16u\_C1R  
image\_mean, [1575](#)
- nppiMeanGetBufferHostSize\_16u\_C3CMR  
image\_mean, [1575](#)
- nppiMeanGetBufferHostSize\_16u\_C3R  
image\_mean, [1575](#)
- nppiMeanGetBufferHostSize\_16u\_C4R  
image\_mean, [1576](#)
- nppiMeanGetBufferHostSize\_32f\_AC4R  
image\_mean, [1576](#)
- nppiMeanGetBufferHostSize\_32f\_C1MR  
image\_mean, [1576](#)
- nppiMeanGetBufferHostSize\_32f\_C1R  
image\_mean, [1576](#)
- nppiMeanGetBufferHostSize\_32f\_C3CMR  
image\_mean, [1577](#)
- nppiMeanGetBufferHostSize\_32f\_C3R  
image\_mean, [1577](#)
- nppiMeanGetBufferHostSize\_32f\_C4R  
image\_mean, [1577](#)
- nppiMeanGetBufferHostSize\_8s\_C1MR  
image\_mean, [1578](#)
- nppiMeanGetBufferHostSize\_8s\_C3CMR  
image\_mean, [1578](#)
- nppiMeanGetBufferHostSize\_8u\_AC4R  
image\_mean, [1578](#)
- nppiMeanGetBufferHostSize\_8u\_C1MR  
image\_mean, [1578](#)
- nppiMeanGetBufferHostSize\_8u\_C1R  
image\_mean, [1579](#)
- nppiMeanGetBufferHostSize\_8u\_C3CMR  
image\_mean, [1579](#)
- nppiMeanGetBufferHostSize\_8u\_C3R  
image\_mean, [1579](#)
- nppiMeanGetBufferHostSize\_8u\_C4R  
image\_mean, [1580](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_C1MR  
image\_mean\_stddev, [1592](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_C1R  
image\_mean\_stddev, [1592](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_C3CMR  
image\_mean\_stddev, [1592](#)
- nppiMeanStdDevGetBufferHostSize\_16u\_C3CR  
image\_mean\_stddev, [1593](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_C1MR  
image\_mean\_stddev, [1593](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_C1R  
image\_mean\_stddev, [1593](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_C3CMR  
image\_mean\_stddev, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_32f\_C3CR  
image\_mean\_stddev, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_C1MR  
image\_mean\_stddev, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_C1R  
image\_mean\_stddev, [1594](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_C3CMR  
image\_mean\_stddev, [1595](#)
- nppiMeanStdDevGetBufferHostSize\_8s\_C3CR  
image\_mean\_stddev, [1595](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_C1MR  
image\_mean\_stddev, [1595](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_C1R  
image\_mean\_stddev, [1596](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_C3CMR  
image\_mean\_stddev, [1596](#)
- nppiMeanStdDevGetBufferHostSize\_8u\_C3CR  
image\_mean\_stddev, [1596](#)
- nppiMin\_16s\_AC4R  
image\_min, [1477](#)
- nppiMin\_16s\_C1R  
image\_min, [1477](#)
- nppiMin\_16s\_C3R  
image\_min, [1478](#)
- nppiMin\_16s\_C4R  
image\_min, [1478](#)
- nppiMin\_16u\_AC4R  
image\_min, [1478](#)



- nppiMin\_16u\_C1R
  - image\_min, [1479](#)
- nppiMin\_16u\_C3R
  - image\_min, [1479](#)
- nppiMin\_16u\_C4R
  - image\_min, [1480](#)
- nppiMin\_32f\_AC4R
  - image\_min, [1480](#)
- nppiMin\_32f\_C1R
  - image\_min, [1480](#)
- nppiMin\_32f\_C3R
  - image\_min, [1481](#)
- nppiMin\_32f\_C4R
  - image\_min, [1481](#)
- nppiMin\_8u\_AC4R
  - image\_min, [1481](#)
- nppiMin\_8u\_C1R
  - image\_min, [1482](#)
- nppiMin\_8u\_C3R
  - image\_min, [1482](#)
- nppiMin\_8u\_C4R
  - image\_min, [1483](#)
- nppiMinEvery\_16s\_AC4IR
  - image\_minevery, [1840](#)
- nppiMinEvery\_16s\_C1IR
  - image\_minevery, [1840](#)
- nppiMinEvery\_16s\_C3IR
  - image\_minevery, [1841](#)
- nppiMinEvery\_16s\_C4IR
  - image\_minevery, [1841](#)
- nppiMinEvery\_16u\_AC4IR
  - image\_minevery, [1841](#)
- nppiMinEvery\_16u\_C1IR
  - image\_minevery, [1842](#)
- nppiMinEvery\_16u\_C3IR
  - image\_minevery, [1842](#)
- nppiMinEvery\_16u\_C4IR
  - image\_minevery, [1842](#)
- nppiMinEvery\_32f\_AC4IR
  - image\_minevery, [1843](#)
- nppiMinEvery\_32f\_C1IR
  - image\_minevery, [1843](#)
- nppiMinEvery\_32f\_C3IR
  - image\_minevery, [1843](#)
- nppiMinEvery\_32f\_C4IR
  - image\_minevery, [1844](#)
- nppiMinEvery\_8u\_AC4IR
  - image\_minevery, [1844](#)
- nppiMinEvery\_8u\_C1IR
  - image\_minevery, [1844](#)
- nppiMinEvery\_8u\_C3IR
  - image\_minevery, [1845](#)
- nppiMinEvery\_8u\_C4IR
  - image\_minevery, [1845](#)
- nppiMinGetBufferHostSize\_16s\_AC4R
  - image\_min, [1483](#)
- nppiMinGetBufferHostSize\_16s\_C1R
  - image\_min, [1483](#)
- nppiMinGetBufferHostSize\_16s\_C3R
  - image\_min, [1483](#)
- nppiMinGetBufferHostSize\_16s\_C4R
  - image\_min, [1484](#)
- nppiMinGetBufferHostSize\_16u\_AC4R
  - image\_min, [1484](#)
- nppiMinGetBufferHostSize\_16u\_C1R
  - image\_min, [1484](#)
- nppiMinGetBufferHostSize\_16u\_C3R
  - image\_min, [1485](#)
- nppiMinGetBufferHostSize\_16u\_C4R
  - image\_min, [1485](#)
- nppiMinGetBufferHostSize\_32f\_AC4R
  - image\_min, [1485](#)
- nppiMinGetBufferHostSize\_32f\_C1R
  - image\_min, [1485](#)
- nppiMinGetBufferHostSize\_32f\_C3R
  - image\_min, [1486](#)
- nppiMinGetBufferHostSize\_32f\_C4R
  - image\_min, [1486](#)
- nppiMinGetBufferHostSize\_8u\_AC4R
  - image\_min, [1486](#)
- nppiMinGetBufferHostSize\_8u\_C1R
  - image\_min, [1487](#)
- nppiMinGetBufferHostSize\_8u\_C3R
  - image\_min, [1487](#)
- nppiMinGetBufferHostSize\_8u\_C4R
  - image\_min, [1487](#)
- nppiMinIndx\_16s\_AC4R
  - image\_min\_index, [1490](#)
- nppiMinIndx\_16s\_C1R
  - image\_min\_index, [1491](#)
- nppiMinIndx\_16s\_C3R
  - image\_min\_index, [1491](#)
- nppiMinIndx\_16s\_C4R
  - image\_min\_index, [1491](#)
- nppiMinIndx\_16u\_AC4R
  - image\_min\_index, [1492](#)
- nppiMinIndx\_16u\_C1R
  - image\_min\_index, [1492](#)
- nppiMinIndx\_16u\_C3R
  - image\_min\_index, [1493](#)
- nppiMinIndx\_16u\_C4R
  - image\_min\_index, [1493](#)
- nppiMinIndx\_32f\_AC4R
  - image\_min\_index, [1493](#)
- nppiMinIndx\_32f\_C1R
  - image\_min\_index, [1494](#)
- nppiMinIndx\_32f\_C3R
  - image\_min\_index, [1494](#)

- nppiMinIndx\_32f\_C4R
  - image\_min\_index, [1495](#)
- nppiMinIndx\_8u\_AC4R
  - image\_min\_index, [1495](#)
- nppiMinIndx\_8u\_C1R
  - image\_min\_index, [1495](#)
- nppiMinIndx\_8u\_C3R
  - image\_min\_index, [1496](#)
- nppiMinIndx\_8u\_C4R
  - image\_min\_index, [1496](#)
- nppiMinIndxGetBufferHostSize\_16s\_AC4R
  - image\_min\_index, [1497](#)
- nppiMinIndxGetBufferHostSize\_16s\_C1R
  - image\_min\_index, [1497](#)
- nppiMinIndxGetBufferHostSize\_16s\_C3R
  - image\_min\_index, [1497](#)
- nppiMinIndxGetBufferHostSize\_16s\_C4R
  - image\_min\_index, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_AC4R
  - image\_min\_index, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_C1R
  - image\_min\_index, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_C3R
  - image\_min\_index, [1498](#)
- nppiMinIndxGetBufferHostSize\_16u\_C4R
  - image\_min\_index, [1499](#)
- nppiMinIndxGetBufferHostSize\_32f\_AC4R
  - image\_min\_index, [1499](#)
- nppiMinIndxGetBufferHostSize\_32f\_C1R
  - image\_min\_index, [1499](#)
- nppiMinIndxGetBufferHostSize\_32f\_C3R
  - image\_min\_index, [1500](#)
- nppiMinIndxGetBufferHostSize\_32f\_C4R
  - image\_min\_index, [1500](#)
- nppiMinIndxGetBufferHostSize\_8u\_AC4R
  - image\_min\_index, [1500](#)
- nppiMinIndxGetBufferHostSize\_8u\_C1R
  - image\_min\_index, [1500](#)
- nppiMinIndxGetBufferHostSize\_8u\_C3R
  - image\_min\_index, [1501](#)
- nppiMinIndxGetBufferHostSize\_8u\_C4R
  - image\_min\_index, [1501](#)
- nppiMinMax\_16s\_AC4R
  - image\_min\_max, [1531](#)
- nppiMinMax\_16s\_C1R
  - image\_min\_max, [1531](#)
- nppiMinMax\_16s\_C3R
  - image\_min\_max, [1532](#)
- nppiMinMax\_16s\_C4R
  - image\_min\_max, [1532](#)
- nppiMinMax\_16u\_AC4R
  - image\_min\_max, [1533](#)
- nppiMinMax\_16u\_C1R
  - image\_min\_max, [1533](#)
- nppiMinMax\_16u\_C3R
  - image\_min\_max, [1533](#)
- nppiMinMax\_16u\_C4R
  - image\_min\_max, [1533](#)
- nppiMinMax\_32f\_AC4R
  - image\_min\_max, [1534](#)
- nppiMinMax\_32f\_C1R
  - image\_min\_max, [1535](#)
- nppiMinMax\_32f\_C3R
  - image\_min\_max, [1535](#)
- nppiMinMax\_32f\_C4R
  - image\_min\_max, [1535](#)
- nppiMinMax\_8u\_AC4R
  - image\_min\_max, [1536](#)
- nppiMinMax\_8u\_C1R
  - image\_min\_max, [1536](#)
- nppiMinMax\_8u\_C3R
  - image\_min\_max, [1537](#)
- nppiMinMax\_8u\_C4R
  - image\_min\_max, [1537](#)
- nppiMinMaxGetBufferHostSize\_16s\_AC4R
  - image\_min\_max, [1537](#)
- nppiMinMaxGetBufferHostSize\_16s\_C1R
  - image\_min\_max, [1538](#)
- nppiMinMaxGetBufferHostSize\_16s\_C3R
  - image\_min\_max, [1538](#)
- nppiMinMaxGetBufferHostSize\_16s\_C4R
  - image\_min\_max, [1538](#)
- nppiMinMaxGetBufferHostSize\_16u\_AC4R
  - image\_min\_max, [1539](#)
- nppiMinMaxGetBufferHostSize\_16u\_C1R
  - image\_min\_max, [1539](#)
- nppiMinMaxGetBufferHostSize\_16u\_C3R
  - image\_min\_max, [1539](#)
- nppiMinMaxGetBufferHostSize\_16u\_C4R
  - image\_min\_max, [1539](#)
- nppiMinMaxGetBufferHostSize\_32f\_AC4R
  - image\_min\_max, [1540](#)
- nppiMinMaxGetBufferHostSize\_32f\_C1R
  - image\_min\_max, [1540](#)
- nppiMinMaxGetBufferHostSize\_32f\_C3R
  - image\_min\_max, [1540](#)
- nppiMinMaxGetBufferHostSize\_32f\_C4R
  - image\_min\_max, [1541](#)
- nppiMinMaxGetBufferHostSize\_8u\_AC4R
  - image\_min\_max, [1541](#)
- nppiMinMaxGetBufferHostSize\_8u\_C1R
  - image\_min\_max, [1541](#)
- nppiMinMaxGetBufferHostSize\_8u\_C3R
  - image\_min\_max, [1541](#)
- nppiMinMaxGetBufferHostSize\_8u\_C4R
  - image\_min\_max, [1542](#)
- nppiMinMaxIndx\_16u\_C1MR
  - image\_min\_max\_index, [1546](#)

- nppiMinMaxIdx\_16u\_C1R
  - image\_min\_max\_index, [1547](#)
- nppiMinMaxIdx\_16u\_C3CMR
  - image\_min\_max\_index, [1547](#)
- nppiMinMaxIdx\_16u\_C3CR
  - image\_min\_max\_index, [1548](#)
- nppiMinMaxIdx\_32f\_C1MR
  - image\_min\_max\_index, [1548](#)
- nppiMinMaxIdx\_32f\_C1R
  - image\_min\_max\_index, [1549](#)
- nppiMinMaxIdx\_32f\_C3CMR
  - image\_min\_max\_index, [1549](#)
- nppiMinMaxIdx\_32f\_C3CR
  - image\_min\_max\_index, [1550](#)
- nppiMinMaxIdx\_8s\_C1MR
  - image\_min\_max\_index, [1551](#)
- nppiMinMaxIdx\_8s\_C1R
  - image\_min\_max\_index, [1551](#)
- nppiMinMaxIdx\_8s\_C3CMR
  - image\_min\_max\_index, [1552](#)
- nppiMinMaxIdx\_8s\_C3CR
  - image\_min\_max\_index, [1552](#)
- nppiMinMaxIdx\_8u\_C1MR
  - image\_min\_max\_index, [1553](#)
- nppiMinMaxIdx\_8u\_C1R
  - image\_min\_max\_index, [1553](#)
- nppiMinMaxIdx\_8u\_C3CMR
  - image\_min\_max\_index, [1554](#)
- nppiMinMaxIdx\_8u\_C3CR
  - image\_min\_max\_index, [1554](#)
- nppiMinMaxIdxGetBufferHostSize\_16u\_C1MR
  - image\_min\_max\_index, [1555](#)
- nppiMinMaxIdxGetBufferHostSize\_16u\_C1R
  - image\_min\_max\_index, [1555](#)
- nppiMinMaxIdxGetBufferHostSize\_16u\_C3CMR
  - image\_min\_max\_index, [1555](#)
- nppiMinMaxIdxGetBufferHostSize\_16u\_C3CR
  - image\_min\_max\_index, [1556](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_C1MR
  - image\_min\_max\_index, [1556](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_C1R
  - image\_min\_max\_index, [1556](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_C3CMR
  - image\_min\_max\_index, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_32f\_C3CR
  - image\_min\_max\_index, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_C1MR
  - image\_min\_max\_index, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_C1R
  - image\_min\_max\_index, [1557](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_C3CMR
  - image\_min\_max\_index, [1558](#)
- nppiMinMaxIdxGetBufferHostSize\_8s\_C3CR
  - image\_min\_max\_index, [1558](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_C1MR
  - image\_min\_max\_index, [1558](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_C1R
  - image\_min\_max\_index, [1559](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_C3CMR
  - image\_min\_max\_index, [1559](#)
- nppiMinMaxIdxGetBufferHostSize\_8u\_C3CR
  - image\_min\_max\_index, [1559](#)
- nppiMirror\_16s\_AC4IR
  - image\_mirror, [1253](#)
- nppiMirror\_16s\_AC4R
  - image\_mirror, [1253](#)
- nppiMirror\_16s\_C1IR
  - image\_mirror, [1254](#)
- nppiMirror\_16s\_C1R
  - image\_mirror, [1254](#)
- nppiMirror\_16s\_C3IR
  - image\_mirror, [1254](#)
- nppiMirror\_16s\_C3R
  - image\_mirror, [1255](#)
- nppiMirror\_16s\_C4IR
  - image\_mirror, [1255](#)
- nppiMirror\_16s\_C4R
  - image\_mirror, [1255](#)
- nppiMirror\_16u\_AC4IR
  - image\_mirror, [1256](#)
- nppiMirror\_16u\_AC4R
  - image\_mirror, [1256](#)
- nppiMirror\_16u\_C1IR
  - image\_mirror, [1256](#)
- nppiMirror\_16u\_C1R
  - image\_mirror, [1257](#)
- nppiMirror\_16u\_C3IR
  - image\_mirror, [1257](#)
- nppiMirror\_16u\_C3R
  - image\_mirror, [1257](#)
- nppiMirror\_16u\_C4IR
  - image\_mirror, [1258](#)
- nppiMirror\_16u\_C4R
  - image\_mirror, [1258](#)
- nppiMirror\_32f\_AC4IR
  - image\_mirror, [1258](#)
- nppiMirror\_32f\_AC4R
  - image\_mirror, [1259](#)
- nppiMirror\_32f\_C1IR
  - image\_mirror, [1259](#)
- nppiMirror\_32f\_C1R
  - image\_mirror, [1259](#)
- nppiMirror\_32f\_C3IR
  - image\_mirror, [1260](#)
- nppiMirror\_32f\_C3R
  - image\_mirror, [1260](#)
- nppiMirror\_32f\_C4IR
  - image\_mirror, [1260](#)

- nppiMirror\_32f\_C4R
  - image\_mirror, [1261](#)
- nppiMirror\_32s\_AC4IR
  - image\_mirror, [1261](#)
- nppiMirror\_32s\_AC4R
  - image\_mirror, [1261](#)
- nppiMirror\_32s\_C1IR
  - image\_mirror, [1262](#)
- nppiMirror\_32s\_C1R
  - image\_mirror, [1262](#)
- nppiMirror\_32s\_C3IR
  - image\_mirror, [1262](#)
- nppiMirror\_32s\_C3R
  - image\_mirror, [1263](#)
- nppiMirror\_32s\_C4IR
  - image\_mirror, [1263](#)
- nppiMirror\_32s\_C4R
  - image\_mirror, [1263](#)
- nppiMirror\_8u\_AC4IR
  - image\_mirror, [1264](#)
- nppiMirror\_8u\_AC4R
  - image\_mirror, [1264](#)
- nppiMirror\_8u\_C1IR
  - image\_mirror, [1264](#)
- nppiMirror\_8u\_C1R
  - image\_mirror, [1265](#)
- nppiMirror\_8u\_C3IR
  - image\_mirror, [1265](#)
- nppiMirror\_8u\_C3R
  - image\_mirror, [1265](#)
- nppiMirror\_8u\_C4IR
  - image\_mirror, [1266](#)
- nppiMirror\_8u\_C4R
  - image\_mirror, [1266](#)
- nppiMul\_16s\_AC4IRSfs
  - image\_mul, [213](#)
- nppiMul\_16s\_AC4RSfs
  - image\_mul, [213](#)
- nppiMul\_16s\_C1IRSfs
  - image\_mul, [214](#)
- nppiMul\_16s\_C1RSfs
  - image\_mul, [214](#)
- nppiMul\_16s\_C3IRSfs
  - image\_mul, [215](#)
- nppiMul\_16s\_C3RSfs
  - image\_mul, [215](#)
- nppiMul\_16s\_C4IRSfs
  - image\_mul, [215](#)
- nppiMul\_16s\_C4RSfs
  - image\_mul, [216](#)
- nppiMul\_16sc\_AC4IRSfs
  - image\_mul, [216](#)
- nppiMul\_16sc\_AC4RSfs
  - image\_mul, [217](#)
- nppiMul\_16sc\_C1IRSfs
  - image\_mul, [217](#)
- nppiMul\_16sc\_C1RSfs
  - image\_mul, [217](#)
- nppiMul\_16sc\_C3IRSfs
  - image\_mul, [218](#)
- nppiMul\_16sc\_C3RSfs
  - image\_mul, [218](#)
- nppiMul\_16u\_AC4IRSfs
  - image\_mul, [219](#)
- nppiMul\_16u\_AC4RSfs
  - image\_mul, [219](#)
- nppiMul\_16u\_C1IRSfs
  - image\_mul, [220](#)
- nppiMul\_16u\_C1RSfs
  - image\_mul, [220](#)
- nppiMul\_16u\_C3IRSfs
  - image\_mul, [220](#)
- nppiMul\_16u\_C3RSfs
  - image\_mul, [221](#)
- nppiMul\_16u\_C4IRSfs
  - image\_mul, [221](#)
- nppiMul\_16u\_C4RSfs
  - image\_mul, [222](#)
- nppiMul\_32f\_AC4IR
  - image\_mul, [222](#)
- nppiMul\_32f\_AC4R
  - image\_mul, [222](#)
- nppiMul\_32f\_C1IR
  - image\_mul, [223](#)
- nppiMul\_32f\_C1R
  - image\_mul, [223](#)
- nppiMul\_32f\_C3IR
  - image\_mul, [224](#)
- nppiMul\_32f\_C3R
  - image\_mul, [224](#)
- nppiMul\_32f\_C4IR
  - image\_mul, [224](#)
- nppiMul\_32f\_C4R
  - image\_mul, [225](#)
- nppiMul\_32fc\_AC4IR
  - image\_mul, [225](#)
- nppiMul\_32fc\_AC4R
  - image\_mul, [225](#)
- nppiMul\_32fc\_C1IR
  - image\_mul, [226](#)
- nppiMul\_32fc\_C1R
  - image\_mul, [226](#)
- nppiMul\_32fc\_C3IR
  - image\_mul, [227](#)
- nppiMul\_32fc\_C3R
  - image\_mul, [227](#)
- nppiMul\_32fc\_C4IR
  - image\_mul, [227](#)

- nppiMul\_32fc\_C4R
  - image\_mul, [228](#)
- nppiMul\_32s\_C1IRSfs
  - image\_mul, [228](#)
- nppiMul\_32s\_C1R
  - image\_mul, [229](#)
- nppiMul\_32s\_C1RSfs
  - image\_mul, [229](#)
- nppiMul\_32s\_C3IRSfs
  - image\_mul, [229](#)
- nppiMul\_32s\_C3RSfs
  - image\_mul, [230](#)
- nppiMul\_32sc\_AC4IRSfs
  - image\_mul, [230](#)
- nppiMul\_32sc\_AC4RSfs
  - image\_mul, [231](#)
- nppiMul\_32sc\_C1IRSfs
  - image\_mul, [231](#)
- nppiMul\_32sc\_C1RSfs
  - image\_mul, [231](#)
- nppiMul\_32sc\_C3IRSfs
  - image\_mul, [232](#)
- nppiMul\_32sc\_C3RSfs
  - image\_mul, [232](#)
- nppiMul\_8u\_AC4IRSfs
  - image\_mul, [233](#)
- nppiMul\_8u\_AC4RSfs
  - image\_mul, [233](#)
- nppiMul\_8u\_C1IRSfs
  - image\_mul, [234](#)
- nppiMul\_8u\_C1RSfs
  - image\_mul, [234](#)
- nppiMul\_8u\_C3IRSfs
  - image\_mul, [234](#)
- nppiMul\_8u\_C3RSfs
  - image\_mul, [235](#)
- nppiMul\_8u\_C4IRSfs
  - image\_mul, [235](#)
- nppiMul\_8u\_C4RSfs
  - image\_mul, [236](#)
- nppiMulC\_16s\_AC4IRSfs
  - image\_mulc, [86](#)
- nppiMulC\_16s\_AC4RSfs
  - image\_mulc, [86](#)
- nppiMulC\_16s\_C1IRSfs
  - image\_mulc, [86](#)
- nppiMulC\_16s\_C1RSfs
  - image\_mulc, [87](#)
- nppiMulC\_16s\_C3IRSfs
  - image\_mulc, [87](#)
- nppiMulC\_16s\_C3RSfs
  - image\_mulc, [87](#)
- nppiMulC\_16s\_C4IRSfs
  - image\_mulc, [88](#)
- nppiMulC\_16s\_C4RSfs
  - image\_mulc, [88](#)
- nppiMulC\_16sc\_AC4IRSfs
  - image\_mulc, [89](#)
- nppiMulC\_16sc\_AC4RSfs
  - image\_mulc, [89](#)
- nppiMulC\_16sc\_C1IRSfs
  - image\_mulc, [89](#)
- nppiMulC\_16sc\_C1RSfs
  - image\_mulc, [90](#)
- nppiMulC\_16sc\_C3IRSfs
  - image\_mulc, [90](#)
- nppiMulC\_16sc\_C3RSfs
  - image\_mulc, [91](#)
- nppiMulC\_16u\_AC4IRSfs
  - image\_mulc, [91](#)
- nppiMulC\_16u\_AC4RSfs
  - image\_mulc, [91](#)
- nppiMulC\_16u\_C1IRSfs
  - image\_mulc, [92](#)
- nppiMulC\_16u\_C1RSfs
  - image\_mulc, [92](#)
- nppiMulC\_16u\_C3IRSfs
  - image\_mulc, [93](#)
- nppiMulC\_16u\_C3RSfs
  - image\_mulc, [93](#)
- nppiMulC\_16u\_C4IRSfs
  - image\_mulc, [93](#)
- nppiMulC\_16u\_C4RSfs
  - image\_mulc, [94](#)
- nppiMulC\_32f\_AC4IR
  - image\_mulc, [94](#)
- nppiMulC\_32f\_AC4R
  - image\_mulc, [94](#)
- nppiMulC\_32f\_C1IR
  - image\_mulc, [95](#)
- nppiMulC\_32f\_C1R
  - image\_mulc, [95](#)
- nppiMulC\_32f\_C3IR
  - image\_mulc, [95](#)
- nppiMulC\_32f\_C3R
  - image\_mulc, [96](#)
- nppiMulC\_32f\_C4IR
  - image\_mulc, [96](#)
- nppiMulC\_32f\_C4R
  - image\_mulc, [96](#)
- nppiMulC\_32fc\_AC4IR
  - image\_mulc, [97](#)
- nppiMulC\_32fc\_AC4R
  - image\_mulc, [97](#)
- nppiMulC\_32fc\_C1IR
  - image\_mulc, [97](#)
- nppiMulC\_32fc\_C1R
  - image\_mulc, [98](#)

nppiMulC\_32fc\_C3IR  
     image\_mulc, [98](#)  
 nppiMulC\_32fc\_C3R  
     image\_mulc, [98](#)  
 nppiMulC\_32fc\_C4IR  
     image\_mulc, [99](#)  
 nppiMulC\_32fc\_C4R  
     image\_mulc, [99](#)  
 nppiMulC\_32s\_C1IRSfs  
     image\_mulc, [100](#)  
 nppiMulC\_32s\_C1RSfs  
     image\_mulc, [100](#)  
 nppiMulC\_32s\_C3IRSfs  
     image\_mulc, [100](#)  
 nppiMulC\_32s\_C3RSfs  
     image\_mulc, [101](#)  
 nppiMulC\_32sc\_AC4IRSfs  
     image\_mulc, [101](#)  
 nppiMulC\_32sc\_AC4RSfs  
     image\_mulc, [101](#)  
 nppiMulC\_32sc\_C1IRSfs  
     image\_mulc, [102](#)  
 nppiMulC\_32sc\_C1RSfs  
     image\_mulc, [102](#)  
 nppiMulC\_32sc\_C3IRSfs  
     image\_mulc, [103](#)  
 nppiMulC\_32sc\_C3RSfs  
     image\_mulc, [103](#)  
 nppiMulC\_8u\_AC4IRSfs  
     image\_mulc, [103](#)  
 nppiMulC\_8u\_AC4RSfs  
     image\_mulc, [104](#)  
 nppiMulC\_8u\_C1IRSfs  
     image\_mulc, [104](#)  
 nppiMulC\_8u\_C1RSfs  
     image\_mulc, [105](#)  
 nppiMulC\_8u\_C3IRSfs  
     image\_mulc, [105](#)  
 nppiMulC\_8u\_C3RSfs  
     image\_mulc, [105](#)  
 nppiMulC\_8u\_C4IRSfs  
     image\_mulc, [106](#)  
 nppiMulC\_8u\_C4RSfs  
     image\_mulc, [106](#)  
 nppiMulCScale\_16u\_AC4IR  
     image\_mulcscale, [108](#)  
 nppiMulCScale\_16u\_AC4R  
     image\_mulcscale, [108](#)  
 nppiMulCScale\_16u\_C1IR  
     image\_mulcscale, [109](#)  
 nppiMulCScale\_16u\_C1R  
     image\_mulcscale, [109](#)  
 nppiMulCScale\_16u\_C3IR  
     image\_mulcscale, [109](#)

nppiMulCScale\_16u\_C3R  
     image\_mulcscale, [110](#)  
 nppiMulCScale\_16u\_C4IR  
     image\_mulcscale, [110](#)  
 nppiMulCScale\_16u\_C4R  
     image\_mulcscale, [110](#)  
 nppiMulCScale\_8u\_AC4IR  
     image\_mulcscale, [111](#)  
 nppiMulCScale\_8u\_AC4R  
     image\_mulcscale, [111](#)  
 nppiMulCScale\_8u\_C1IR  
     image\_mulcscale, [111](#)  
 nppiMulCScale\_8u\_C1R  
     image\_mulcscale, [112](#)  
 nppiMulCScale\_8u\_C3IR  
     image\_mulcscale, [112](#)  
 nppiMulCScale\_8u\_C3R  
     image\_mulcscale, [112](#)  
 nppiMulCScale\_8u\_C4IR  
     image\_mulcscale, [113](#)  
 nppiMulCScale\_8u\_C4R  
     image\_mulcscale, [113](#)  
 nppiMulScale\_16u\_AC4IR  
     image\_mulscale, [238](#)  
 nppiMulScale\_16u\_AC4R  
     image\_mulscale, [239](#)  
 nppiMulScale\_16u\_C1IR  
     image\_mulscale, [239](#)  
 nppiMulScale\_16u\_C1R  
     image\_mulscale, [239](#)  
 nppiMulScale\_16u\_C3IR  
     image\_mulscale, [240](#)  
 nppiMulScale\_16u\_C3R  
     image\_mulscale, [240](#)  
 nppiMulScale\_16u\_C4IR  
     image\_mulscale, [241](#)  
 nppiMulScale\_16u\_C4R  
     image\_mulscale, [241](#)  
 nppiMulScale\_8u\_AC4IR  
     image\_mulscale, [241](#)  
 nppiMulScale\_8u\_AC4R  
     image\_mulscale, [242](#)  
 nppiMulScale\_8u\_C1IR  
     image\_mulscale, [242](#)  
 nppiMulScale\_8u\_C1R  
     image\_mulscale, [243](#)  
 nppiMulScale\_8u\_C3IR  
     image\_mulscale, [243](#)  
 nppiMulScale\_8u\_C3R  
     image\_mulscale, [243](#)  
 nppiMulScale\_8u\_C4IR  
     image\_mulscale, [244](#)  
 nppiMulScale\_8u\_C4R  
     image\_mulscale, [244](#)

- nppiNorm\_Inf\_16s\_AC4R
  - image\_inf\_norm, [1603](#)
- nppiNorm\_Inf\_16s\_C1R
  - image\_inf\_norm, [1603](#)
- nppiNorm\_Inf\_16s\_C3R
  - image\_inf\_norm, [1603](#)
- nppiNorm\_Inf\_16s\_C4R
  - image\_inf\_norm, [1604](#)
- nppiNorm\_Inf\_16u\_AC4R
  - image\_inf\_norm, [1604](#)
- nppiNorm\_Inf\_16u\_C1MR
  - image\_inf\_norm, [1604](#)
- nppiNorm\_Inf\_16u\_C1R
  - image\_inf\_norm, [1605](#)
- nppiNorm\_Inf\_16u\_C3CMR
  - image\_inf\_norm, [1605](#)
- nppiNorm\_Inf\_16u\_C3R
  - image\_inf\_norm, [1606](#)
- nppiNorm\_Inf\_16u\_C4R
  - image\_inf\_norm, [1606](#)
- nppiNorm\_Inf\_32f\_AC4R
  - image\_inf\_norm, [1606](#)
- nppiNorm\_Inf\_32f\_C1MR
  - image\_inf\_norm, [1607](#)
- nppiNorm\_Inf\_32f\_C1R
  - image\_inf\_norm, [1607](#)
- nppiNorm\_Inf\_32f\_C3CMR
  - image\_inf\_norm, [1608](#)
- nppiNorm\_Inf\_32f\_C3R
  - image\_inf\_norm, [1608](#)
- nppiNorm\_Inf\_32f\_C4R
  - image\_inf\_norm, [1608](#)
- nppiNorm\_Inf\_32s\_C1R
  - image\_inf\_norm, [1609](#)
- nppiNorm\_Inf\_8s\_C1MR
  - image\_inf\_norm, [1609](#)
- nppiNorm\_Inf\_8s\_C3CMR
  - image\_inf\_norm, [1610](#)
- nppiNorm\_Inf\_8u\_AC4R
  - image\_inf\_norm, [1610](#)
- nppiNorm\_Inf\_8u\_C1MR
  - image\_inf\_norm, [1610](#)
- nppiNorm\_Inf\_8u\_C1R
  - image\_inf\_norm, [1611](#)
- nppiNorm\_Inf\_8u\_C3CMR
  - image\_inf\_norm, [1611](#)
- nppiNorm\_Inf\_8u\_C3R
  - image\_inf\_norm, [1612](#)
- nppiNorm\_Inf\_8u\_C4R
  - image\_inf\_norm, [1612](#)
- nppiNorm\_L1\_16s\_AC4R
  - image\_L1\_norm, [1625](#)
- nppiNorm\_L1\_16s\_C1R
  - image\_L1\_norm, [1625](#)
- nppiNorm\_L1\_16s\_C3R
  - image\_L1\_norm, [1625](#)
- nppiNorm\_L1\_16s\_C4R
  - image\_L1\_norm, [1626](#)
- nppiNorm\_L1\_16u\_AC4R
  - image\_L1\_norm, [1626](#)
- nppiNorm\_L1\_16u\_C1MR
  - image\_L1\_norm, [1626](#)
- nppiNorm\_L1\_16u\_C1R
  - image\_L1\_norm, [1627](#)
- nppiNorm\_L1\_16u\_C3CMR
  - image\_L1\_norm, [1627](#)
- nppiNorm\_L1\_16u\_C3R
  - image\_L1\_norm, [1628](#)
- nppiNorm\_L1\_16u\_C4R
  - image\_L1\_norm, [1628](#)
- nppiNorm\_L1\_32f\_AC4R
  - image\_L1\_norm, [1628](#)
- nppiNorm\_L1\_32f\_C1MR
  - image\_L1\_norm, [1629](#)
- nppiNorm\_L1\_32f\_C1R
  - image\_L1\_norm, [1629](#)
- nppiNorm\_L1\_32f\_C3CMR
  - image\_L1\_norm, [1629](#)
- nppiNorm\_L1\_32f\_C3R
  - image\_L1\_norm, [1630](#)
- nppiNorm\_L1\_32f\_C4R
  - image\_L1\_norm, [1630](#)
- nppiNorm\_L1\_8s\_C1MR
  - image\_L1\_norm, [1631](#)
- nppiNorm\_L1\_8s\_C3CMR
  - image\_L1\_norm, [1631](#)
- nppiNorm\_L1\_8u\_AC4R
  - image\_L1\_norm, [1631](#)
- nppiNorm\_L1\_8u\_C1MR
  - image\_L1\_norm, [1632](#)
- nppiNorm\_L1\_8u\_C1R
  - image\_L1\_norm, [1632](#)
- nppiNorm\_L1\_8u\_C3CMR
  - image\_L1\_norm, [1633](#)
- nppiNorm\_L1\_8u\_C3R
  - image\_L1\_norm, [1633](#)
- nppiNorm\_L1\_8u\_C4R
  - image\_L1\_norm, [1633](#)
- nppiNorm\_L2\_16s\_AC4R
  - image\_L2\_norm, [1646](#)
- nppiNorm\_L2\_16s\_C1R
  - image\_L2\_norm, [1646](#)
- nppiNorm\_L2\_16s\_C3R
  - image\_L2\_norm, [1646](#)
- nppiNorm\_L2\_16s\_C4R
  - image\_L2\_norm, [1647](#)
- nppiNorm\_L2\_16u\_AC4R
  - image\_L2\_norm, [1647](#)



- nppiNorm\_L2\_16u\_C1MR  
image\_L2\_norm, [1647](#)
- nppiNorm\_L2\_16u\_C1R  
image\_L2\_norm, [1648](#)
- nppiNorm\_L2\_16u\_C3CMR  
image\_L2\_norm, [1648](#)
- nppiNorm\_L2\_16u\_C3R  
image\_L2\_norm, [1649](#)
- nppiNorm\_L2\_16u\_C4R  
image\_L2\_norm, [1649](#)
- nppiNorm\_L2\_32f\_AC4R  
image\_L2\_norm, [1649](#)
- nppiNorm\_L2\_32f\_C1MR  
image\_L2\_norm, [1650](#)
- nppiNorm\_L2\_32f\_C1R  
image\_L2\_norm, [1650](#)
- nppiNorm\_L2\_32f\_C3CMR  
image\_L2\_norm, [1650](#)
- nppiNorm\_L2\_32f\_C3R  
image\_L2\_norm, [1651](#)
- nppiNorm\_L2\_32f\_C4R  
image\_L2\_norm, [1651](#)
- nppiNorm\_L2\_8s\_C1MR  
image\_L2\_norm, [1652](#)
- nppiNorm\_L2\_8s\_C3CMR  
image\_L2\_norm, [1652](#)
- nppiNorm\_L2\_8u\_AC4R  
image\_L2\_norm, [1652](#)
- nppiNorm\_L2\_8u\_C1MR  
image\_L2\_norm, [1653](#)
- nppiNorm\_L2\_8u\_C1R  
image\_L2\_norm, [1653](#)
- nppiNorm\_L2\_8u\_C3CMR  
image\_L2\_norm, [1654](#)
- nppiNorm\_L2\_8u\_C3R  
image\_L2\_norm, [1654](#)
- nppiNorm\_L2\_8u\_C4R  
image\_L2\_norm, [1654](#)
- nppiNormDiff\_Inf\_16s\_AC4R  
image\_inf\_normdiff, [1667](#)
- nppiNormDiff\_Inf\_16s\_C1R  
image\_inf\_normdiff, [1667](#)
- nppiNormDiff\_Inf\_16s\_C3R  
image\_inf\_normdiff, [1668](#)
- nppiNormDiff\_Inf\_16s\_C4R  
image\_inf\_normdiff, [1668](#)
- nppiNormDiff\_Inf\_16u\_AC4R  
image\_inf\_normdiff, [1669](#)
- nppiNormDiff\_Inf\_16u\_C1MR  
image\_inf\_normdiff, [1669](#)
- nppiNormDiff\_Inf\_16u\_C1R  
image\_inf\_normdiff, [1670](#)
- nppiNormDiff\_Inf\_16u\_C3CMR  
image\_inf\_normdiff, [1670](#)
- nppiNormDiff\_Inf\_16u\_C3R  
image\_inf\_normdiff, [1671](#)
- nppiNormDiff\_Inf\_16u\_C4R  
image\_inf\_normdiff, [1671](#)
- nppiNormDiff\_Inf\_32f\_AC4R  
image\_inf\_normdiff, [1671](#)
- nppiNormDiff\_Inf\_32f\_C1MR  
image\_inf\_normdiff, [1672](#)
- nppiNormDiff\_Inf\_32f\_C1R  
image\_inf\_normdiff, [1672](#)
- nppiNormDiff\_Inf\_32f\_C3CMR  
image\_inf\_normdiff, [1673](#)
- nppiNormDiff\_Inf\_32f\_C3R  
image\_inf\_normdiff, [1673](#)
- nppiNormDiff\_Inf\_32f\_C4R  
image\_inf\_normdiff, [1674](#)
- nppiNormDiff\_Inf\_8s\_C1MR  
image\_inf\_normdiff, [1674](#)
- nppiNormDiff\_Inf\_8s\_C3CMR  
image\_inf\_normdiff, [1675](#)
- nppiNormDiff\_Inf\_8u\_AC4R  
image\_inf\_normdiff, [1675](#)
- nppiNormDiff\_Inf\_8u\_C1MR  
image\_inf\_normdiff, [1676](#)
- nppiNormDiff\_Inf\_8u\_C1R  
image\_inf\_normdiff, [1676](#)
- nppiNormDiff\_Inf\_8u\_C3CMR  
image\_inf\_normdiff, [1677](#)
- nppiNormDiff\_Inf\_8u\_C3R  
image\_inf\_normdiff, [1677](#)
- nppiNormDiff\_Inf\_8u\_C4R  
image\_inf\_normdiff, [1678](#)
- nppiNormDiff\_L1\_16s\_AC4R  
image\_L1\_normdiff, [1690](#)
- nppiNormDiff\_L1\_16s\_C1R  
image\_L1\_normdiff, [1690](#)
- nppiNormDiff\_L1\_16s\_C3R  
image\_L1\_normdiff, [1691](#)
- nppiNormDiff\_L1\_16s\_C4R  
image\_L1\_normdiff, [1691](#)
- nppiNormDiff\_L1\_16u\_AC4R  
image\_L1\_normdiff, [1692](#)
- nppiNormDiff\_L1\_16u\_C1MR  
image\_L1\_normdiff, [1692](#)
- nppiNormDiff\_L1\_16u\_C1R  
image\_L1\_normdiff, [1692](#)
- nppiNormDiff\_L1\_16u\_C3CMR  
image\_L1\_normdiff, [1693](#)
- nppiNormDiff\_L1\_16u\_C3R  
image\_L1\_normdiff, [1693](#)
- nppiNormDiff\_L1\_16u\_C4R  
image\_L1\_normdiff, [1694](#)
- nppiNormDiff\_L1\_32f\_AC4R  
image\_L1\_normdiff, [1694](#)



- nppiNormDiff\_L1\_32f\_C1MR  
image\_L1\_normdiff, [1695](#)
- nppiNormDiff\_L1\_32f\_C1R  
image\_L1\_normdiff, [1695](#)
- nppiNormDiff\_L1\_32f\_C3CMR  
image\_L1\_normdiff, [1696](#)
- nppiNormDiff\_L1\_32f\_C3R  
image\_L1\_normdiff, [1696](#)
- nppiNormDiff\_L1\_32f\_C4R  
image\_L1\_normdiff, [1697](#)
- nppiNormDiff\_L1\_8s\_C1MR  
image\_L1\_normdiff, [1697](#)
- nppiNormDiff\_L1\_8s\_C3CMR  
image\_L1\_normdiff, [1698](#)
- nppiNormDiff\_L1\_8u\_AC4R  
image\_L1\_normdiff, [1698](#)
- nppiNormDiff\_L1\_8u\_C1MR  
image\_L1\_normdiff, [1699](#)
- nppiNormDiff\_L1\_8u\_C1R  
image\_L1\_normdiff, [1699](#)
- nppiNormDiff\_L1\_8u\_C3CMR  
image\_L1\_normdiff, [1699](#)
- nppiNormDiff\_L1\_8u\_C3R  
image\_L1\_normdiff, [1700](#)
- nppiNormDiff\_L1\_8u\_C4R  
image\_L1\_normdiff, [1700](#)
- nppiNormDiff\_L2\_16s\_AC4R  
image\_L2\_normdiff, [1713](#)
- nppiNormDiff\_L2\_16s\_C1R  
image\_L2\_normdiff, [1713](#)
- nppiNormDiff\_L2\_16s\_C3R  
image\_L2\_normdiff, [1714](#)
- nppiNormDiff\_L2\_16s\_C4R  
image\_L2\_normdiff, [1714](#)
- nppiNormDiff\_L2\_16u\_AC4R  
image\_L2\_normdiff, [1715](#)
- nppiNormDiff\_L2\_16u\_C1MR  
image\_L2\_normdiff, [1715](#)
- nppiNormDiff\_L2\_16u\_C1R  
image\_L2\_normdiff, [1715](#)
- nppiNormDiff\_L2\_16u\_C3CMR  
image\_L2\_normdiff, [1716](#)
- nppiNormDiff\_L2\_16u\_C3R  
image\_L2\_normdiff, [1716](#)
- nppiNormDiff\_L2\_16u\_C4R  
image\_L2\_normdiff, [1717](#)
- nppiNormDiff\_L2\_32f\_AC4R  
image\_L2\_normdiff, [1717](#)
- nppiNormDiff\_L2\_32f\_C1MR  
image\_L2\_normdiff, [1718](#)
- nppiNormDiff\_L2\_32f\_C1R  
image\_L2\_normdiff, [1718](#)
- nppiNormDiff\_L2\_32f\_C3CMR  
image\_L2\_normdiff, [1719](#)
- nppiNormDiff\_L2\_32f\_C3R  
image\_L2\_normdiff, [1719](#)
- nppiNormDiff\_L2\_32f\_C4R  
image\_L2\_normdiff, [1720](#)
- nppiNormDiff\_L2\_8s\_C1MR  
image\_L2\_normdiff, [1720](#)
- nppiNormDiff\_L2\_8s\_C3CMR  
image\_L2\_normdiff, [1721](#)
- nppiNormDiff\_L2\_8u\_AC4R  
image\_L2\_normdiff, [1721](#)
- nppiNormDiff\_L2\_8u\_C1MR  
image\_L2\_normdiff, [1722](#)
- nppiNormDiff\_L2\_8u\_C1R  
image\_L2\_normdiff, [1722](#)
- nppiNormDiff\_L2\_8u\_C3CMR  
image\_L2\_normdiff, [1722](#)
- nppiNormDiff\_L2\_8u\_C3R  
image\_L2\_normdiff, [1723](#)
- nppiNormDiff\_L2\_8u\_C4R  
image\_L2\_normdiff, [1723](#)
- nppiNormDiffInfGetBufferHostSize\_16s\_AC4R  
image\_inf\_normdiff, [1678](#)
- nppiNormDiffInfGetBufferHostSize\_16s\_C1R  
image\_inf\_normdiff, [1678](#)
- nppiNormDiffInfGetBufferHostSize\_16s\_C3R  
image\_inf\_normdiff, [1679](#)
- nppiNormDiffInfGetBufferHostSize\_16s\_C4R  
image\_inf\_normdiff, [1679](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_AC4R  
image\_inf\_normdiff, [1679](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_C1MR  
image\_inf\_normdiff, [1680](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_C1R  
image\_inf\_normdiff, [1680](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_C3CMR  
image\_inf\_normdiff, [1680](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_C3R  
image\_inf\_normdiff, [1680](#)
- nppiNormDiffInfGetBufferHostSize\_16u\_C4R  
image\_inf\_normdiff, [1681](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_AC4R  
image\_inf\_normdiff, [1681](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_C1MR  
image\_inf\_normdiff, [1681](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_C1R  
image\_inf\_normdiff, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_C3CMR  
image\_inf\_normdiff, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_C3R  
image\_inf\_normdiff, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_32f\_C4R  
image\_inf\_normdiff, [1682](#)
- nppiNormDiffInfGetBufferHostSize\_8s\_C1MR  
image\_inf\_normdiff, [1683](#)

- nppiNormDiffInfGetBufferHostSize\_8s\_C3CMR  
image\_inf\_normdiff, [1683](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_AC4R  
image\_inf\_normdiff, [1683](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C1MR  
image\_inf\_normdiff, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C1R  
image\_inf\_normdiff, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C3CMR  
image\_inf\_normdiff, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C3R  
image\_inf\_normdiff, [1684](#)
- nppiNormDiffInfGetBufferHostSize\_8u\_C4R  
image\_inf\_normdiff, [1685](#)
- nppiNormDiffL1GetBufferHostSize\_16s\_AC4R  
image\_L1\_normdiff, [1701](#)
- nppiNormDiffL1GetBufferHostSize\_16s\_C1R  
image\_L1\_normdiff, [1701](#)
- nppiNormDiffL1GetBufferHostSize\_16s\_C3R  
image\_L1\_normdiff, [1701](#)
- nppiNormDiffL1GetBufferHostSize\_16s\_C4R  
image\_L1\_normdiff, [1702](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_AC4R  
image\_L1\_normdiff, [1702](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_C1MR  
image\_L1\_normdiff, [1702](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_C1R  
image\_L1\_normdiff, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_C3CMR  
image\_L1\_normdiff, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_C3R  
image\_L1\_normdiff, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_16u\_C4R  
image\_L1\_normdiff, [1703](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_AC4R  
image\_L1\_normdiff, [1704](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_C1MR  
image\_L1\_normdiff, [1704](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_C1R  
image\_L1\_normdiff, [1704](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_C3CMR  
image\_L1\_normdiff, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_C3R  
image\_L1\_normdiff, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_32f\_C4R  
image\_L1\_normdiff, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_8s\_C1MR  
image\_L1\_normdiff, [1705](#)
- nppiNormDiffL1GetBufferHostSize\_8s\_C3CMR  
image\_L1\_normdiff, [1706](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_AC4R  
image\_L1\_normdiff, [1706](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C1MR  
image\_L1\_normdiff, [1706](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C1R  
image\_L1\_normdiff, [1707](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C3CMR  
image\_L1\_normdiff, [1707](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C3R  
image\_L1\_normdiff, [1707](#)
- nppiNormDiffL1GetBufferHostSize\_8u\_C4R  
image\_L1\_normdiff, [1707](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_AC4R  
image\_L2\_normdiff, [1724](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_C1R  
image\_L2\_normdiff, [1724](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_C3R  
image\_L2\_normdiff, [1724](#)
- nppiNormDiffL2GetBufferHostSize\_16s\_C4R  
image\_L2\_normdiff, [1725](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_AC4R  
image\_L2\_normdiff, [1725](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_C1MR  
image\_L2\_normdiff, [1725](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_C1R  
image\_L2\_normdiff, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_C3CMR  
image\_L2\_normdiff, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_C3R  
image\_L2\_normdiff, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_16u\_C4R  
image\_L2\_normdiff, [1726](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_AC4R  
image\_L2\_normdiff, [1727](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_C1MR  
image\_L2\_normdiff, [1727](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_C1R  
image\_L2\_normdiff, [1727](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_C3CMR  
image\_L2\_normdiff, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_C3R  
image\_L2\_normdiff, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_32f\_C4R  
image\_L2\_normdiff, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_8s\_C1MR  
image\_L2\_normdiff, [1728](#)
- nppiNormDiffL2GetBufferHostSize\_8s\_C3CMR  
image\_L2\_normdiff, [1729](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_AC4R  
image\_L2\_normdiff, [1729](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_C1MR  
image\_L2\_normdiff, [1729](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_C1R  
image\_L2\_normdiff, [1730](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_C3CMR  
image\_L2\_normdiff, [1730](#)
- nppiNormDiffL2GetBufferHostSize\_8u\_C3R  
image\_L2\_normdiff, [1730](#)

- nppiNormDiffL2GetBufferHostSize\_8u\_C4R  
image\_L2\_normdiff, 1730
- nppiNormInfGetBufferHostSize\_16s\_AC4R  
image\_inf\_norm, 1612
- nppiNormInfGetBufferHostSize\_16s\_C1R  
image\_inf\_norm, 1613
- nppiNormInfGetBufferHostSize\_16s\_C3R  
image\_inf\_norm, 1613
- nppiNormInfGetBufferHostSize\_16s\_C4R  
image\_inf\_norm, 1613
- nppiNormInfGetBufferHostSize\_16u\_AC4R  
image\_inf\_norm, 1614
- nppiNormInfGetBufferHostSize\_16u\_C1MR  
image\_inf\_norm, 1614
- nppiNormInfGetBufferHostSize\_16u\_C1R  
image\_inf\_norm, 1614
- nppiNormInfGetBufferHostSize\_16u\_C3CMR  
image\_inf\_norm, 1614
- nppiNormInfGetBufferHostSize\_16u\_C3R  
image\_inf\_norm, 1615
- nppiNormInfGetBufferHostSize\_16u\_C4R  
image\_inf\_norm, 1615
- nppiNormInfGetBufferHostSize\_32f\_AC4R  
image\_inf\_norm, 1615
- nppiNormInfGetBufferHostSize\_32f\_C1MR  
image\_inf\_norm, 1616
- nppiNormInfGetBufferHostSize\_32f\_C1R  
image\_inf\_norm, 1616
- nppiNormInfGetBufferHostSize\_32f\_C3CMR  
image\_inf\_norm, 1616
- nppiNormInfGetBufferHostSize\_32f\_C3R  
image\_inf\_norm, 1616
- nppiNormInfGetBufferHostSize\_32f\_C4R  
image\_inf\_norm, 1617
- nppiNormInfGetBufferHostSize\_32s\_C1R  
image\_inf\_norm, 1617
- nppiNormInfGetBufferHostSize\_8s\_C1MR  
image\_inf\_norm, 1617
- nppiNormInfGetBufferHostSize\_8s\_C3CMR  
image\_inf\_norm, 1618
- nppiNormInfGetBufferHostSize\_8u\_AC4R  
image\_inf\_norm, 1618
- nppiNormInfGetBufferHostSize\_8u\_C1MR  
image\_inf\_norm, 1618
- nppiNormInfGetBufferHostSize\_8u\_C1R  
image\_inf\_norm, 1618
- nppiNormInfGetBufferHostSize\_8u\_C3CMR  
image\_inf\_norm, 1619
- nppiNormInfGetBufferHostSize\_8u\_C3R  
image\_inf\_norm, 1619
- nppiNormInfGetBufferHostSize\_8u\_C4R  
image\_inf\_norm, 1619
- nppiNormL1GetBufferHostSize\_16s\_AC4R  
image\_L1\_norm, 1634
- nppiNormL1GetBufferHostSize\_16s\_C1R  
image\_L1\_norm, 1634
- nppiNormL1GetBufferHostSize\_16s\_C3R  
image\_L1\_norm, 1634
- nppiNormL1GetBufferHostSize\_16s\_C4R  
image\_L1\_norm, 1635
- nppiNormL1GetBufferHostSize\_16u\_AC4R  
image\_L1\_norm, 1635
- nppiNormL1GetBufferHostSize\_16u\_C1MR  
image\_L1\_norm, 1635
- nppiNormL1GetBufferHostSize\_16u\_C1R  
image\_L1\_norm, 1636
- nppiNormL1GetBufferHostSize\_16u\_C3CMR  
image\_L1\_norm, 1636
- nppiNormL1GetBufferHostSize\_16u\_C3R  
image\_L1\_norm, 1636
- nppiNormL1GetBufferHostSize\_16u\_C4R  
image\_L1\_norm, 1636
- nppiNormL1GetBufferHostSize\_32f\_AC4R  
image\_L1\_norm, 1637
- nppiNormL1GetBufferHostSize\_32f\_C1MR  
image\_L1\_norm, 1637
- nppiNormL1GetBufferHostSize\_32f\_C1R  
image\_L1\_norm, 1637
- nppiNormL1GetBufferHostSize\_32f\_C3CMR  
image\_L1\_norm, 1638
- nppiNormL1GetBufferHostSize\_32f\_C3R  
image\_L1\_norm, 1638
- nppiNormL1GetBufferHostSize\_32f\_C4R  
image\_L1\_norm, 1638
- nppiNormL1GetBufferHostSize\_8s\_C1MR  
image\_L1\_norm, 1638
- nppiNormL1GetBufferHostSize\_8s\_C3CMR  
image\_L1\_norm, 1639
- nppiNormL1GetBufferHostSize\_8u\_AC4R  
image\_L1\_norm, 1639
- nppiNormL1GetBufferHostSize\_8u\_C1MR  
image\_L1\_norm, 1639
- nppiNormL1GetBufferHostSize\_8u\_C1R  
image\_L1\_norm, 1640
- nppiNormL1GetBufferHostSize\_8u\_C3CMR  
image\_L1\_norm, 1640
- nppiNormL1GetBufferHostSize\_8u\_C3R  
image\_L1\_norm, 1640
- nppiNormL1GetBufferHostSize\_8u\_C4R  
image\_L1\_norm, 1640
- nppiNormL2GetBufferHostSize\_16s\_AC4R  
image\_L2\_norm, 1655
- nppiNormL2GetBufferHostSize\_16s\_C1R  
image\_L2\_norm, 1655
- nppiNormL2GetBufferHostSize\_16s\_C3R  
image\_L2\_norm, 1655
- nppiNormL2GetBufferHostSize\_16s\_C4R  
image\_L2\_norm, 1656

- nppiNormL2GetBufferHostSize\_16u\_AC4R  
image\_L2\_norm, [1656](#)
- nppiNormL2GetBufferHostSize\_16u\_C1MR  
image\_L2\_norm, [1656](#)
- nppiNormL2GetBufferHostSize\_16u\_C1R  
image\_L2\_norm, [1657](#)
- nppiNormL2GetBufferHostSize\_16u\_C3CMR  
image\_L2\_norm, [1657](#)
- nppiNormL2GetBufferHostSize\_16u\_C3R  
image\_L2\_norm, [1657](#)
- nppiNormL2GetBufferHostSize\_16u\_C4R  
image\_L2\_norm, [1657](#)
- nppiNormL2GetBufferHostSize\_32f\_AC4R  
image\_L2\_norm, [1658](#)
- nppiNormL2GetBufferHostSize\_32f\_C1MR  
image\_L2\_norm, [1658](#)
- nppiNormL2GetBufferHostSize\_32f\_C1R  
image\_L2\_norm, [1658](#)
- nppiNormL2GetBufferHostSize\_32f\_C3CMR  
image\_L2\_norm, [1659](#)
- nppiNormL2GetBufferHostSize\_32f\_C3R  
image\_L2\_norm, [1659](#)
- nppiNormL2GetBufferHostSize\_32f\_C4R  
image\_L2\_norm, [1659](#)
- nppiNormL2GetBufferHostSize\_8s\_C1MR  
image\_L2\_norm, [1659](#)
- nppiNormL2GetBufferHostSize\_8s\_C3CMR  
image\_L2\_norm, [1660](#)
- nppiNormL2GetBufferHostSize\_8u\_AC4R  
image\_L2\_norm, [1660](#)
- nppiNormL2GetBufferHostSize\_8u\_C1MR  
image\_L2\_norm, [1660](#)
- nppiNormL2GetBufferHostSize\_8u\_C1R  
image\_L2\_norm, [1661](#)
- nppiNormL2GetBufferHostSize\_8u\_C3CMR  
image\_L2\_norm, [1661](#)
- nppiNormL2GetBufferHostSize\_8u\_C3R  
image\_L2\_norm, [1661](#)
- nppiNormL2GetBufferHostSize\_8u\_C4R  
image\_L2\_norm, [1661](#)
- nppiNormRel\_Inf\_16s\_AC4R  
image\_inf\_normrel, [1736](#)
- nppiNormRel\_Inf\_16s\_C1R  
image\_inf\_normrel, [1736](#)
- nppiNormRel\_Inf\_16s\_C3R  
image\_inf\_normrel, [1737](#)
- nppiNormRel\_Inf\_16s\_C4R  
image\_inf\_normrel, [1737](#)
- nppiNormRel\_Inf\_16u\_AC4R  
image\_inf\_normrel, [1738](#)
- nppiNormRel\_Inf\_16u\_C1MR  
image\_inf\_normrel, [1738](#)
- nppiNormRel\_Inf\_16u\_C1R  
image\_inf\_normrel, [1739](#)
- nppiNormRel\_Inf\_16u\_C3CMR  
image\_inf\_normrel, [1739](#)
- nppiNormRel\_Inf\_16u\_C3R  
image\_inf\_normrel, [1740](#)
- nppiNormRel\_Inf\_16u\_C4R  
image\_inf\_normrel, [1740](#)
- nppiNormRel\_Inf\_32f\_AC4R  
image\_inf\_normrel, [1740](#)
- nppiNormRel\_Inf\_32f\_C1MR  
image\_inf\_normrel, [1741](#)
- nppiNormRel\_Inf\_32f\_C1R  
image\_inf\_normrel, [1741](#)
- nppiNormRel\_Inf\_32f\_C3CMR  
image\_inf\_normrel, [1742](#)
- nppiNormRel\_Inf\_32f\_C3R  
image\_inf\_normrel, [1742](#)
- nppiNormRel\_Inf\_32f\_C4R  
image\_inf\_normrel, [1743](#)
- nppiNormRel\_Inf\_8s\_C1MR  
image\_inf\_normrel, [1743](#)
- nppiNormRel\_Inf\_8s\_C3CMR  
image\_inf\_normrel, [1744](#)
- nppiNormRel\_Inf\_8u\_AC4R  
image\_inf\_normrel, [1744](#)
- nppiNormRel\_Inf\_8u\_C1MR  
image\_inf\_normrel, [1745](#)
- nppiNormRel\_Inf\_8u\_C1R  
image\_inf\_normrel, [1745](#)
- nppiNormRel\_Inf\_8u\_C3CMR  
image\_inf\_normrel, [1746](#)
- nppiNormRel\_Inf\_8u\_C3R  
image\_inf\_normrel, [1746](#)
- nppiNormRel\_Inf\_8u\_C4R  
image\_inf\_normrel, [1747](#)
- nppiNormRel\_L1\_16s\_AC4R  
image\_L1\_normrel, [1759](#)
- nppiNormRel\_L1\_16s\_C1R  
image\_L1\_normrel, [1759](#)
- nppiNormRel\_L1\_16s\_C3R  
image\_L1\_normrel, [1760](#)
- nppiNormRel\_L1\_16s\_C4R  
image\_L1\_normrel, [1760](#)
- nppiNormRel\_L1\_16u\_AC4R  
image\_L1\_normrel, [1761](#)
- nppiNormRel\_L1\_16u\_C1MR  
image\_L1\_normrel, [1761](#)
- nppiNormRel\_L1\_16u\_C1R  
image\_L1\_normrel, [1762](#)
- nppiNormRel\_L1\_16u\_C3CMR  
image\_L1\_normrel, [1762](#)
- nppiNormRel\_L1\_16u\_C3R  
image\_L1\_normrel, [1762](#)
- nppiNormRel\_L1\_16u\_C4R  
image\_L1\_normrel, [1763](#)

- nppiNormRel\_L1\_32f\_AC4R
  - image\_L1\_normrel, [1763](#)
- nppiNormRel\_L1\_32f\_C1MR
  - image\_L1\_normrel, [1764](#)
- nppiNormRel\_L1\_32f\_C1R
  - image\_L1\_normrel, [1764](#)
- nppiNormRel\_L1\_32f\_C3CMR
  - image\_L1\_normrel, [1765](#)
- nppiNormRel\_L1\_32f\_C3R
  - image\_L1\_normrel, [1765](#)
- nppiNormRel\_L1\_32f\_C4R
  - image\_L1\_normrel, [1766](#)
- nppiNormRel\_L1\_8s\_C1MR
  - image\_L1\_normrel, [1766](#)
- nppiNormRel\_L1\_8s\_C3CMR
  - image\_L1\_normrel, [1767](#)
- nppiNormRel\_L1\_8u\_AC4R
  - image\_L1\_normrel, [1767](#)
- nppiNormRel\_L1\_8u\_C1MR
  - image\_L1\_normrel, [1768](#)
- nppiNormRel\_L1\_8u\_C1R
  - image\_L1\_normrel, [1768](#)
- nppiNormRel\_L1\_8u\_C3CMR
  - image\_L1\_normrel, [1769](#)
- nppiNormRel\_L1\_8u\_C3R
  - image\_L1\_normrel, [1769](#)
- nppiNormRel\_L1\_8u\_C4R
  - image\_L1\_normrel, [1770](#)
- nppiNormRel\_L2\_16s\_AC4R
  - image\_L2\_normrel, [1782](#)
- nppiNormRel\_L2\_16s\_C1R
  - image\_L2\_normrel, [1782](#)
- nppiNormRel\_L2\_16s\_C3R
  - image\_L2\_normrel, [1783](#)
- nppiNormRel\_L2\_16s\_C4R
  - image\_L2\_normrel, [1783](#)
- nppiNormRel\_L2\_16u\_AC4R
  - image\_L2\_normrel, [1784](#)
- nppiNormRel\_L2\_16u\_C1MR
  - image\_L2\_normrel, [1784](#)
- nppiNormRel\_L2\_16u\_C1R
  - image\_L2\_normrel, [1785](#)
- nppiNormRel\_L2\_16u\_C3CMR
  - image\_L2\_normrel, [1785](#)
- nppiNormRel\_L2\_16u\_C3R
  - image\_L2\_normrel, [1785](#)
- nppiNormRel\_L2\_16u\_C4R
  - image\_L2\_normrel, [1786](#)
- nppiNormRel\_L2\_32f\_AC4R
  - image\_L2\_normrel, [1786](#)
- nppiNormRel\_L2\_32f\_C1MR
  - image\_L2\_normrel, [1787](#)
- nppiNormRel\_L2\_32f\_C1R
  - image\_L2\_normrel, [1787](#)
- nppiNormRel\_L2\_32f\_C3CMR
  - image\_L2\_normrel, [1788](#)
- nppiNormRel\_L2\_32f\_C3R
  - image\_L2\_normrel, [1788](#)
- nppiNormRel\_L2\_32f\_C4R
  - image\_L2\_normrel, [1789](#)
- nppiNormRel\_L2\_8s\_C1MR
  - image\_L2\_normrel, [1789](#)
- nppiNormRel\_L2\_8s\_C3CMR
  - image\_L2\_normrel, [1790](#)
- nppiNormRel\_L2\_8u\_AC4R
  - image\_L2\_normrel, [1790](#)
- nppiNormRel\_L2\_8u\_C1MR
  - image\_L2\_normrel, [1791](#)
- nppiNormRel\_L2\_8u\_C1R
  - image\_L2\_normrel, [1791](#)
- nppiNormRel\_L2\_8u\_C3CMR
  - image\_L2\_normrel, [1792](#)
- nppiNormRel\_L2\_8u\_C3R
  - image\_L2\_normrel, [1792](#)
- nppiNormRel\_L2\_8u\_C4R
  - image\_L2\_normrel, [1793](#)
- nppiNormRelInfGetBufferHostSize\_16s\_AC4R
  - image\_inf\_normrel, [1747](#)
- nppiNormRelInfGetBufferHostSize\_16s\_C1R
  - image\_inf\_normrel, [1748](#)
- nppiNormRelInfGetBufferHostSize\_16s\_C3R
  - image\_inf\_normrel, [1748](#)
- nppiNormRelInfGetBufferHostSize\_16s\_C4R
  - image\_inf\_normrel, [1748](#)
- nppiNormRelInfGetBufferHostSize\_16u\_AC4R
  - image\_inf\_normrel, [1748](#)
- nppiNormRelInfGetBufferHostSize\_16u\_C1MR
  - image\_inf\_normrel, [1749](#)
- nppiNormRelInfGetBufferHostSize\_16u\_C1R
  - image\_inf\_normrel, [1749](#)
- nppiNormRelInfGetBufferHostSize\_16u\_C3CMR
  - image\_inf\_normrel, [1749](#)
- nppiNormRelInfGetBufferHostSize\_16u\_C3R
  - image\_inf\_normrel, [1750](#)
- nppiNormRelInfGetBufferHostSize\_16u\_C4R
  - image\_inf\_normrel, [1750](#)
- nppiNormRelInfGetBufferHostSize\_32f\_AC4R
  - image\_inf\_normrel, [1750](#)
- nppiNormRelInfGetBufferHostSize\_32f\_C1MR
  - image\_inf\_normrel, [1750](#)
- nppiNormRelInfGetBufferHostSize\_32f\_C1R
  - image\_inf\_normrel, [1751](#)
- nppiNormRelInfGetBufferHostSize\_32f\_C3CMR
  - image\_inf\_normrel, [1751](#)
- nppiNormRelInfGetBufferHostSize\_32f\_C3R
  - image\_inf\_normrel, [1751](#)
- nppiNormRelInfGetBufferHostSize\_32f\_C4R
  - image\_inf\_normrel, [1752](#)



- nppiNormRelInfGetBufferHostSize\_32s\_C1R  
image\_inf\_normrel, [1752](#)
- nppiNormRelInfGetBufferHostSize\_8s\_C1MR  
image\_inf\_normrel, [1752](#)
- nppiNormRelInfGetBufferHostSize\_8s\_C3CMR  
image\_inf\_normrel, [1752](#)
- nppiNormRelInfGetBufferHostSize\_8u\_AC4R  
image\_inf\_normrel, [1753](#)
- nppiNormRelInfGetBufferHostSize\_8u\_C1MR  
image\_inf\_normrel, [1753](#)
- nppiNormRelInfGetBufferHostSize\_8u\_C1R  
image\_inf\_normrel, [1753](#)
- nppiNormRelInfGetBufferHostSize\_8u\_C3CMR  
image\_inf\_normrel, [1754](#)
- nppiNormRelInfGetBufferHostSize\_8u\_C3R  
image\_inf\_normrel, [1754](#)
- nppiNormRelInfGetBufferHostSize\_8u\_C4R  
image\_inf\_normrel, [1754](#)
- nppiNormRelL1GetBufferHostSize\_16s\_AC4R  
image\_L1\_normrel, [1770](#)
- nppiNormRelL1GetBufferHostSize\_16s\_C1R  
image\_L1\_normrel, [1770](#)
- nppiNormRelL1GetBufferHostSize\_16s\_C3R  
image\_L1\_normrel, [1771](#)
- nppiNormRelL1GetBufferHostSize\_16s\_C4R  
image\_L1\_normrel, [1771](#)
- nppiNormRelL1GetBufferHostSize\_16u\_AC4R  
image\_L1\_normrel, [1771](#)
- nppiNormRelL1GetBufferHostSize\_16u\_C1MR  
image\_L1\_normrel, [1772](#)
- nppiNormRelL1GetBufferHostSize\_16u\_C1R  
image\_L1\_normrel, [1772](#)
- nppiNormRelL1GetBufferHostSize\_16u\_C3CMR  
image\_L1\_normrel, [1772](#)
- nppiNormRelL1GetBufferHostSize\_16u\_C3R  
image\_L1\_normrel, [1772](#)
- nppiNormRelL1GetBufferHostSize\_16u\_C4R  
image\_L1\_normrel, [1773](#)
- nppiNormRelL1GetBufferHostSize\_32f\_AC4R  
image\_L1\_normrel, [1773](#)
- nppiNormRelL1GetBufferHostSize\_32f\_C1MR  
image\_L1\_normrel, [1773](#)
- nppiNormRelL1GetBufferHostSize\_32f\_C1R  
image\_L1\_normrel, [1774](#)
- nppiNormRelL1GetBufferHostSize\_32f\_C3CMR  
image\_L1\_normrel, [1774](#)
- nppiNormRelL1GetBufferHostSize\_32f\_C3R  
image\_L1\_normrel, [1774](#)
- nppiNormRelL1GetBufferHostSize\_32f\_C4R  
image\_L1\_normrel, [1774](#)
- nppiNormRelL1GetBufferHostSize\_8s\_C1MR  
image\_L1\_normrel, [1775](#)
- nppiNormRelL1GetBufferHostSize\_8s\_C3CMR  
image\_L1\_normrel, [1775](#)
- nppiNormRelL1GetBufferHostSize\_8u\_AC4R  
image\_L1\_normrel, [1775](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C1MR  
image\_L1\_normrel, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C1R  
image\_L1\_normrel, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C3CMR  
image\_L1\_normrel, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C3R  
image\_L1\_normrel, [1776](#)
- nppiNormRelL1GetBufferHostSize\_8u\_C4R  
image\_L1\_normrel, [1777](#)
- nppiNormRelL2GetBufferHostSize\_16s\_AC4R  
image\_L2\_normrel, [1793](#)
- nppiNormRelL2GetBufferHostSize\_16s\_C1R  
image\_L2\_normrel, [1793](#)
- nppiNormRelL2GetBufferHostSize\_16s\_C3R  
image\_L2\_normrel, [1794](#)
- nppiNormRelL2GetBufferHostSize\_16s\_C4R  
image\_L2\_normrel, [1794](#)
- nppiNormRelL2GetBufferHostSize\_16u\_AC4R  
image\_L2\_normrel, [1794](#)
- nppiNormRelL2GetBufferHostSize\_16u\_C1MR  
image\_L2\_normrel, [1795](#)
- nppiNormRelL2GetBufferHostSize\_16u\_C1R  
image\_L2\_normrel, [1795](#)
- nppiNormRelL2GetBufferHostSize\_16u\_C3CMR  
image\_L2\_normrel, [1795](#)
- nppiNormRelL2GetBufferHostSize\_16u\_C3R  
image\_L2\_normrel, [1795](#)
- nppiNormRelL2GetBufferHostSize\_16u\_C4R  
image\_L2\_normrel, [1796](#)
- nppiNormRelL2GetBufferHostSize\_32f\_AC4R  
image\_L2\_normrel, [1796](#)
- nppiNormRelL2GetBufferHostSize\_32f\_C1MR  
image\_L2\_normrel, [1796](#)
- nppiNormRelL2GetBufferHostSize\_32f\_C1R  
image\_L2\_normrel, [1797](#)
- nppiNormRelL2GetBufferHostSize\_32f\_C3CMR  
image\_L2\_normrel, [1797](#)
- nppiNormRelL2GetBufferHostSize\_32f\_C3R  
image\_L2\_normrel, [1797](#)
- nppiNormRelL2GetBufferHostSize\_32f\_C4R  
image\_L2\_normrel, [1797](#)
- nppiNormRelL2GetBufferHostSize\_8s\_C1MR  
image\_L2\_normrel, [1798](#)
- nppiNormRelL2GetBufferHostSize\_8s\_C3CMR  
image\_L2\_normrel, [1798](#)
- nppiNormRelL2GetBufferHostSize\_8u\_AC4R  
image\_L2\_normrel, [1798](#)
- nppiNormRelL2GetBufferHostSize\_8u\_C1MR  
image\_L2\_normrel, [1799](#)
- nppiNormRelL2GetBufferHostSize\_8u\_C1R  
image\_L2\_normrel, [1799](#)

- nppiNormRelL2GetBufferHostSize\_8u\_C3CMR
  - image\_L2\_normrel, [1799](#)
- nppiNormRelL2GetBufferHostSize\_8u\_C3R
  - image\_L2\_normrel, [1799](#)
- nppiNormRelL2GetBufferHostSize\_8u\_C4R
  - image\_L2\_normrel, [1800](#)
- nppiNot\_8u\_AC4IR
  - image\_not, [468](#)
- nppiNot\_8u\_AC4R
  - image\_not, [469](#)
- nppiNot\_8u\_C1IR
  - image\_not, [469](#)
- nppiNot\_8u\_C1R
  - image\_not, [469](#)
- nppiNot\_8u\_C3IR
  - image\_not, [469](#)
- nppiNot\_8u\_C3R
  - image\_not, [470](#)
- nppiNot\_8u\_C4IR
  - image\_not, [470](#)
- nppiNot\_8u\_C4R
  - image\_not, [470](#)
- nppiOr\_16u\_AC4IR
  - image\_or, [446](#)
- nppiOr\_16u\_AC4R
  - image\_or, [446](#)
- nppiOr\_16u\_C1IR
  - image\_or, [446](#)
- nppiOr\_16u\_C1R
  - image\_or, [447](#)
- nppiOr\_16u\_C3IR
  - image\_or, [447](#)
- nppiOr\_16u\_C3R
  - image\_or, [447](#)
- nppiOr\_16u\_C4IR
  - image\_or, [448](#)
- nppiOr\_16u\_C4R
  - image\_or, [448](#)
- nppiOr\_32s\_AC4IR
  - image\_or, [449](#)
- nppiOr\_32s\_AC4R
  - image\_or, [449](#)
- nppiOr\_32s\_C1IR
  - image\_or, [449](#)
- nppiOr\_32s\_C1R
  - image\_or, [450](#)
- nppiOr\_32s\_C3IR
  - image\_or, [450](#)
- nppiOr\_32s\_C3R
  - image\_or, [450](#)
- nppiOr\_32s\_C4IR
  - image\_or, [451](#)
- nppiOr\_32s\_C4R
  - image\_or, [451](#)
- nppiOr\_8u\_AC4IR
  - image\_or, [452](#)
- nppiOr\_8u\_AC4R
  - image\_or, [452](#)
- nppiOr\_8u\_C1IR
  - image\_or, [452](#)
- nppiOr\_8u\_C1R
  - image\_or, [453](#)
- nppiOr\_8u\_C3IR
  - image\_or, [453](#)
- nppiOr\_8u\_C3R
  - image\_or, [453](#)
- nppiOr\_8u\_C4IR
  - image\_or, [454](#)
- nppiOr\_8u\_C4R
  - image\_or, [454](#)
- nppiOrC\_16u\_AC4IR
  - image\_orc, [384](#)
- nppiOrC\_16u\_AC4R
  - image\_orc, [384](#)
- nppiOrC\_16u\_C1IR
  - image\_orc, [384](#)
- nppiOrC\_16u\_C1R
  - image\_orc, [385](#)
- nppiOrC\_16u\_C3IR
  - image\_orc, [385](#)
- nppiOrC\_16u\_C3R
  - image\_orc, [385](#)
- nppiOrC\_16u\_C4IR
  - image\_orc, [386](#)
- nppiOrC\_16u\_C4R
  - image\_orc, [386](#)
- nppiOrC\_32s\_AC4IR
  - image\_orc, [386](#)
- nppiOrC\_32s\_AC4R
  - image\_orc, [387](#)
- nppiOrC\_32s\_C1IR
  - image\_orc, [387](#)
- nppiOrC\_32s\_C1R
  - image\_orc, [387](#)
- nppiOrC\_32s\_C3IR
  - image\_orc, [388](#)
- nppiOrC\_32s\_C3R
  - image\_orc, [388](#)
- nppiOrC\_32s\_C4IR
  - image\_orc, [388](#)
- nppiOrC\_32s\_C4R
  - image\_orc, [389](#)
- nppiOrC\_8u\_AC4IR
  - image\_orc, [389](#)
- nppiOrC\_8u\_AC4R
  - image\_orc, [389](#)
- nppiOrC\_8u\_C1IR
  - image\_orc, [390](#)

- image\_orc, [390](#)
- nppiOrC\_8u\_C3IR
  - image\_orc, [390](#)
- nppiOrC\_8u\_C3R
  - image\_orc, [391](#)
- nppiOrC\_8u\_C4IR
  - image\_orc, [391](#)
- nppiOrC\_8u\_C4R
  - image\_orc, [391](#)
- NppiPoint, [2631](#)
  - x, [2631](#)
  - y, [2631](#)
- nppiQualityIndex\_16u32f\_AC4R
  - image\_quality\_index, [2017](#)
- nppiQualityIndex\_16u32f\_C1R
  - image\_quality\_index, [2017](#)
- nppiQualityIndex\_16u32f\_C3R
  - image\_quality\_index, [2018](#)
- nppiQualityIndex\_32f\_AC4R
  - image\_quality\_index, [2018](#)
- nppiQualityIndex\_32f\_C1R
  - image\_quality\_index, [2019](#)
- nppiQualityIndex\_32f\_C3R
  - image\_quality\_index, [2019](#)
- nppiQualityIndex\_8u32f\_AC4R
  - image\_quality\_index, [2019](#)
- nppiQualityIndex\_8u32f\_C1R
  - image\_quality\_index, [2020](#)
- nppiQualityIndex\_8u32f\_C3R
  - image\_quality\_index, [2020](#)
- nppiQualityIndexGetBufferHostSize\_16u32f\_-AC4R
  - image\_quality\_index, [2021](#)
- nppiQualityIndexGetBufferHostSize\_16u32f\_C1R
  - image\_quality\_index, [2021](#)
- nppiQualityIndexGetBufferHostSize\_16u32f\_C3R
  - image\_quality\_index, [2021](#)
- nppiQualityIndexGetBufferHostSize\_32f\_AC4R
  - image\_quality\_index, [2022](#)
- nppiQualityIndexGetBufferHostSize\_32f\_C1R
  - image\_quality\_index, [2022](#)
- nppiQualityIndexGetBufferHostSize\_32f\_C3R
  - image\_quality\_index, [2022](#)
- nppiQualityIndexGetBufferHostSize\_8u32f\_AC4R
  - image\_quality\_index, [2023](#)
- nppiQualityIndexGetBufferHostSize\_8u32f\_C1R
  - image\_quality\_index, [2023](#)
- nppiQualityIndexGetBufferHostSize\_8u32f\_C3R
  - image\_quality\_index, [2023](#)
- nppiQuantFwdRawTableInit\_JPEG\_8u
  - image\_quantization, [708](#)
- nppiQuantFwdTableInit\_JPEG\_8u16u
  - image\_quantization, [709](#)
- nppiQuantInvTableInit\_JPEG\_8u16u
  - image\_quantization, [709](#)
- NppiRect, [2632](#)
  - height, [2632](#)
  - width, [2632](#)
  - x, [2632](#)
  - y, [2632](#)
- nppiRectStdDev\_32f\_C1R
  - image\_rectstddev, [1851](#)
- nppiRectStdDev\_32s32f\_C1R
  - image\_rectstddev, [1852](#)
- nppiRectStdDev\_32s\_C1RSfs
  - image\_rectstddev, [1852](#)
- nppiRemap\_16s\_AC4R
  - image\_remap, [1222](#)
- nppiRemap\_16s\_C1R
  - image\_remap, [1223](#)
- nppiRemap\_16s\_C3R
  - image\_remap, [1223](#)
- nppiRemap\_16s\_C4R
  - image\_remap, [1224](#)
- nppiRemap\_16s\_P3R
  - image\_remap, [1225](#)
- nppiRemap\_16s\_P4R
  - image\_remap, [1225](#)
- nppiRemap\_16u\_AC4R
  - image\_remap, [1226](#)
- nppiRemap\_16u\_C1R
  - image\_remap, [1226](#)
- nppiRemap\_16u\_C3R
  - image\_remap, [1227](#)
- nppiRemap\_16u\_C4R
  - image\_remap, [1228](#)
- nppiRemap\_16u\_P3R
  - image\_remap, [1228](#)
- nppiRemap\_16u\_P4R
  - image\_remap, [1229](#)
- nppiRemap\_32f\_AC4R
  - image\_remap, [1229](#)
- nppiRemap\_32f\_C1R
  - image\_remap, [1230](#)
- nppiRemap\_32f\_C3R
  - image\_remap, [1231](#)
- nppiRemap\_32f\_C4R
  - image\_remap, [1231](#)
- nppiRemap\_32f\_P3R
  - image\_remap, [1232](#)
- nppiRemap\_32f\_P4R
  - image\_remap, [1232](#)
- nppiRemap\_64f\_AC4R
  - image\_remap, [1233](#)
- nppiRemap\_64f\_C1R
  - image\_remap, [1234](#)
- nppiRemap\_64f\_C3R



- image\_remap, [1234](#)
- nppiRemap\_64f\_C4R
  - image\_remap, [1235](#)
- nppiRemap\_64f\_P3R
  - image\_remap, [1235](#)
- nppiRemap\_64f\_P4R
  - image\_remap, [1236](#)
- nppiRemap\_8u\_AC4R
  - image\_remap, [1237](#)
- nppiRemap\_8u\_C1R
  - image\_remap, [1237](#)
- nppiRemap\_8u\_C3R
  - image\_remap, [1238](#)
- nppiRemap\_8u\_C4R
  - image\_remap, [1238](#)
- nppiRemap\_8u\_P3R
  - image\_remap, [1239](#)
- nppiRemap\_8u\_P4R
  - image\_remap, [1240](#)
- nppiResize\_16u\_AC4R
  - image\_resize, [1209](#)
- nppiResize\_16u\_C1R
  - image\_resize, [1210](#)
- nppiResize\_16u\_C3R
  - image\_resize, [1210](#)
- nppiResize\_16u\_C4R
  - image\_resize, [1211](#)
- nppiResize\_16u\_P3R
  - image\_resize, [1211](#)
- nppiResize\_16u\_P4R
  - image\_resize, [1212](#)
- nppiResize\_32f\_AC4R
  - image\_resize, [1212](#)
- nppiResize\_32f\_C1R
  - image\_resize, [1213](#)
- nppiResize\_32f\_C3R
  - image\_resize, [1213](#)
- nppiResize\_32f\_C4R
  - image\_resize, [1214](#)
- nppiResize\_32f\_P3R
  - image\_resize, [1214](#)
- nppiResize\_32f\_P4R
  - image\_resize, [1215](#)
- nppiResize\_8u\_AC4R
  - image\_resize, [1215](#)
- nppiResize\_8u\_C1R
  - image\_resize, [1216](#)
- nppiResize\_8u\_C3R
  - image\_resize, [1216](#)
- nppiResize\_8u\_C4R
  - image\_resize, [1217](#)
- nppiResize\_8u\_P3R
  - image\_resize, [1217](#)
- nppiResize\_8u\_P4R
  - image\_resize, [1218](#)
- nppiResizeSqrPixel\_16s\_AC4R
  - image\_resize\_square\_pixel, [1189](#)
- nppiResizeSqrPixel\_16s\_C1R
  - image\_resize\_square\_pixel, [1190](#)
- nppiResizeSqrPixel\_16s\_C3R
  - image\_resize\_square\_pixel, [1190](#)
- nppiResizeSqrPixel\_16s\_C4R
  - image\_resize\_square\_pixel, [1191](#)
- nppiResizeSqrPixel\_16s\_P3R
  - image\_resize\_square\_pixel, [1191](#)
- nppiResizeSqrPixel\_16s\_P4R
  - image\_resize\_square\_pixel, [1192](#)
- nppiResizeSqrPixel\_16u\_AC4R
  - image\_resize\_square\_pixel, [1192](#)
- nppiResizeSqrPixel\_16u\_C1R
  - image\_resize\_square\_pixel, [1193](#)
- nppiResizeSqrPixel\_16u\_C3R
  - image\_resize\_square\_pixel, [1193](#)
- nppiResizeSqrPixel\_16u\_C4R
  - image\_resize\_square\_pixel, [1194](#)
- nppiResizeSqrPixel\_16u\_P3R
  - image\_resize\_square\_pixel, [1194](#)
- nppiResizeSqrPixel\_16u\_P4R
  - image\_resize\_square\_pixel, [1195](#)
- nppiResizeSqrPixel\_32f\_AC4R
  - image\_resize\_square\_pixel, [1196](#)
- nppiResizeSqrPixel\_32f\_C1R
  - image\_resize\_square\_pixel, [1196](#)
- nppiResizeSqrPixel\_32f\_C3R
  - image\_resize\_square\_pixel, [1197](#)
- nppiResizeSqrPixel\_32f\_C4R
  - image\_resize\_square\_pixel, [1197](#)
- nppiResizeSqrPixel\_32f\_P3R
  - image\_resize\_square\_pixel, [1198](#)
- nppiResizeSqrPixel\_32f\_P4R
  - image\_resize\_square\_pixel, [1198](#)
- nppiResizeSqrPixel\_64f\_AC4R
  - image\_resize\_square\_pixel, [1199](#)
- nppiResizeSqrPixel\_64f\_C1R
  - image\_resize\_square\_pixel, [1200](#)
- nppiResizeSqrPixel\_64f\_C3R
  - image\_resize\_square\_pixel, [1200](#)
- nppiResizeSqrPixel\_64f\_C4R
  - image\_resize\_square\_pixel, [1201](#)
- nppiResizeSqrPixel\_64f\_P3R
  - image\_resize\_square\_pixel, [1201](#)
- nppiResizeSqrPixel\_64f\_P4R
  - image\_resize\_square\_pixel, [1202](#)
- nppiResizeSqrPixel\_8u\_AC4R
  - image\_resize\_square\_pixel, [1202](#)
- nppiResizeSqrPixel\_8u\_C1R
  - image\_resize\_square\_pixel, [1203](#)
- nppiResizeSqrPixel\_8u\_C3R

image\_resize\_square\_pixel, 1203  
 nppiResizeSqrPixel\_8u\_C4R  
     image\_resize\_square\_pixel, 1204  
 nppiResizeSqrPixel\_8u\_P3R  
     image\_resize\_square\_pixel, 1204  
 nppiResizeSqrPixel\_8u\_P4R  
     image\_resize\_square\_pixel, 1205  
 nppiRGBToCbYCr422\_8u\_C3C2R  
     image\_color\_model\_conversion, 547  
 nppiRGBToCbYCr422Gamma\_8u\_C3C2R  
     image\_color\_model\_conversion, 547  
 nppiRGBToGray\_16s\_AC4C1R  
     image\_color\_model\_conversion, 548  
 nppiRGBToGray\_16s\_C3C1R  
     image\_color\_model\_conversion, 548  
 nppiRGBToGray\_16u\_AC4C1R  
     image\_color\_model\_conversion, 548  
 nppiRGBToGray\_16u\_C3C1R  
     image\_color\_model\_conversion, 549  
 nppiRGBToGray\_32f\_AC4C1R  
     image\_color\_model\_conversion, 549  
 nppiRGBToGray\_32f\_C3C1R  
     image\_color\_model\_conversion, 549  
 nppiRGBToGray\_8u\_AC4C1R  
     image\_color\_model\_conversion, 550  
 nppiRGBToGray\_8u\_C3C1R  
     image\_color\_model\_conversion, 550  
 nppiRGBToHLS\_8u\_AC4R  
     image\_color\_model\_conversion, 550  
 nppiRGBToHLS\_8u\_C3R  
     image\_color\_model\_conversion, 551  
 nppiRGBToHSV\_8u\_AC4R  
     image\_color\_model\_conversion, 551  
 nppiRGBToHSV\_8u\_C3R  
     image\_color\_model\_conversion, 551  
 nppiRGBToLUV\_8u\_AC4R  
     image\_color\_model\_conversion, 552  
 nppiRGBToLUV\_8u\_C3R  
     image\_color\_model\_conversion, 552  
 nppiRGBToXYZ\_8u\_AC4R  
     image\_color\_model\_conversion, 552  
 nppiRGBToXYZ\_8u\_C3R  
     image\_color\_model\_conversion, 553  
 nppiRGBToYCbCr420\_8u\_C3P3R  
     image\_color\_model\_conversion, 553  
 nppiRGBToYCbCr422\_8u\_C3C2R  
     image\_color\_model\_conversion, 553  
 nppiRGBToYCbCr422\_8u\_C3P3R  
     image\_color\_model\_conversion, 554  
 nppiRGBToYCbCr422\_8u\_P3C2R  
     image\_color\_model\_conversion, 554  
 nppiRGBToYCbCr\_8u\_AC4P3R  
     image\_color\_model\_conversion, 555  
 nppiRGBToYCbCr\_8u\_AC4R

image\_color\_model\_conversion, 555  
 nppiRGBToYCbCr\_8u\_C3P3R  
     image\_color\_model\_conversion, 555  
 nppiRGBToYCbCr\_8u\_C3R  
     image\_color\_model\_conversion, 556  
 nppiRGBToYCbCr\_8u\_P3R  
     image\_color\_model\_conversion, 556  
 nppiRGBToYCC\_8u\_AC4R  
     image\_color\_model\_conversion, 556  
 nppiRGBToYCC\_8u\_C3R  
     image\_color\_model\_conversion, 557  
 nppiRGBToYCrCb420\_8u\_AC4P3R  
     image\_color\_model\_conversion, 557  
 nppiRGBToYCrCb422\_8u\_C3C2R  
     image\_color\_model\_conversion, 557  
 nppiRGBToYCrCb422\_8u\_P3C2R  
     image\_color\_model\_conversion, 558  
 nppiRGBToYUV420\_8u\_C3P3R  
     image\_color\_model\_conversion, 558  
 nppiRGBToYUV420\_8u\_P3R  
     image\_color\_model\_conversion, 558  
 nppiRGBToYUV422\_8u\_C3C2R  
     image\_color\_model\_conversion, 559  
 nppiRGBToYUV422\_8u\_C3P3R  
     image\_color\_model\_conversion, 559  
 nppiRGBToYUV422\_8u\_P3R  
     image\_color\_model\_conversion, 559  
 nppiRGBToYUV\_8u\_AC4P4R  
     image\_color\_model\_conversion, 560  
 nppiRGBToYUV\_8u\_AC4R  
     image\_color\_model\_conversion, 560  
 nppiRGBToYUV\_8u\_C3P3R  
     image\_color\_model\_conversion, 561  
 nppiRGBToYUV\_8u\_C3R  
     image\_color\_model\_conversion, 561  
 nppiRGBToYUV\_8u\_P3R  
     image\_color\_model\_conversion, 561  
 nppiRotate\_16u\_AC4R  
     image\_rotate, 1243  
 nppiRotate\_16u\_C1R  
     image\_rotate, 1244  
 nppiRotate\_16u\_C3R  
     image\_rotate, 1244  
 nppiRotate\_16u\_C4R  
     image\_rotate, 1245  
 nppiRotate\_32f\_AC4R  
     image\_rotate, 1245  
 nppiRotate\_32f\_C1R  
     image\_rotate, 1246  
 nppiRotate\_32f\_C3R  
     image\_rotate, 1246  
 nppiRotate\_32f\_C4R  
     image\_rotate, 1247  
 nppiRotate\_8u\_AC4R

- image\_rotate, [1247](#)
- nppiRotate\_8u\_C1R
  - image\_rotate, [1248](#)
- nppiRotate\_8u\_C3R
  - image\_rotate, [1248](#)
- nppiRotate\_8u\_C4R
  - image\_rotate, [1249](#)
- nppiRShiftC\_16s\_AC4IR
  - image\_rshiftc, [407](#)
- nppiRShiftC\_16s\_AC4R
  - image\_rshiftc, [407](#)
- nppiRShiftC\_16s\_C1IR
  - image\_rshiftc, [408](#)
- nppiRShiftC\_16s\_C1R
  - image\_rshiftc, [408](#)
- nppiRShiftC\_16s\_C3IR
  - image\_rshiftc, [408](#)
- nppiRShiftC\_16s\_C3R
  - image\_rshiftc, [409](#)
- nppiRShiftC\_16s\_C4IR
  - image\_rshiftc, [409](#)
- nppiRShiftC\_16s\_C4R
  - image\_rshiftc, [409](#)
- nppiRShiftC\_16u\_AC4IR
  - image\_rshiftc, [410](#)
- nppiRShiftC\_16u\_AC4R
  - image\_rshiftc, [410](#)
- nppiRShiftC\_16u\_C1IR
  - image\_rshiftc, [410](#)
- nppiRShiftC\_16u\_C1R
  - image\_rshiftc, [411](#)
- nppiRShiftC\_16u\_C3IR
  - image\_rshiftc, [411](#)
- nppiRShiftC\_16u\_C3R
  - image\_rshiftc, [411](#)
- nppiRShiftC\_16u\_C4IR
  - image\_rshiftc, [412](#)
- nppiRShiftC\_16u\_C4R
  - image\_rshiftc, [412](#)
- nppiRShiftC\_32s\_AC4IR
  - image\_rshiftc, [412](#)
- nppiRShiftC\_32s\_AC4R
  - image\_rshiftc, [413](#)
- nppiRShiftC\_32s\_C1IR
  - image\_rshiftc, [413](#)
- nppiRShiftC\_32s\_C1R
  - image\_rshiftc, [413](#)
- nppiRShiftC\_32s\_C3IR
  - image\_rshiftc, [414](#)
- nppiRShiftC\_32s\_C3R
  - image\_rshiftc, [414](#)
- nppiRShiftC\_32s\_C4IR
  - image\_rshiftc, [414](#)
- nppiRShiftC\_32s\_C4R
  - image\_rshiftc, [415](#)
- nppiRShiftC\_8s\_AC4IR
  - image\_rshiftc, [415](#)
- nppiRShiftC\_8s\_AC4R
  - image\_rshiftc, [415](#)
- nppiRShiftC\_8s\_C1IR
  - image\_rshiftc, [416](#)
- nppiRShiftC\_8s\_C1R
  - image\_rshiftc, [416](#)
- nppiRShiftC\_8s\_C3IR
  - image\_rshiftc, [416](#)
- nppiRShiftC\_8s\_C3R
  - image\_rshiftc, [417](#)
- nppiRShiftC\_8s\_C4IR
  - image\_rshiftc, [417](#)
- nppiRShiftC\_8s\_C4R
  - image\_rshiftc, [417](#)
- nppiRShiftC\_8u\_AC4IR
  - image\_rshiftc, [418](#)
- nppiRShiftC\_8u\_AC4R
  - image\_rshiftc, [418](#)
- nppiRShiftC\_8u\_C1IR
  - image\_rshiftc, [418](#)
- nppiRShiftC\_8u\_C1R
  - image\_rshiftc, [419](#)
- nppiRShiftC\_8u\_C3IR
  - image\_rshiftc, [419](#)
- nppiRShiftC\_8u\_C3R
  - image\_rshiftc, [419](#)
- nppiRShiftC\_8u\_C4IR
  - image\_rshiftc, [420](#)
- nppiRShiftC\_8u\_C4R
  - image\_rshiftc, [420](#)
- nppiSameNormLevelGetBufferHostSize\_16u32f\_-AC4R
  - crosscorrsgamenormlevel, [1988](#)
- nppiSameNormLevelGetBufferHostSize\_16u32f\_-C1R
  - crosscorrsgamenormlevel, [1989](#)
- nppiSameNormLevelGetBufferHostSize\_16u32f\_-C3R
  - crosscorrsgamenormlevel, [1989](#)
- nppiSameNormLevelGetBufferHostSize\_16u32f\_-C4R
  - crosscorrsgamenormlevel, [1989](#)
- nppiSameNormLevelGetBufferHostSize\_32f\_-AC4R
  - crosscorrsgamenormlevel, [1990](#)
- nppiSameNormLevelGetBufferHostSize\_32f\_-C1R
  - crosscorrsgamenormlevel, [1990](#)
- nppiSameNormLevelGetBufferHostSize\_32f\_-C3R
  - crosscorrsgamenormlevel, [1990](#)
- nppiSameNormLevelGetBufferHostSize\_32f\_-C4R
  - crosscorrsgamenormlevel, [1990](#)

- AC4R
  - crosscorrmenormlevel, [1991](#)
- nppiSameNormLevelGetBufferHostSize\_8s32f\_-C1R
  - crosscorrmenormlevel, [1991](#)
- nppiSameNormLevelGetBufferHostSize\_8s32f\_-C3R
  - crosscorrmenormlevel, [1991](#)
- nppiSameNormLevelGetBufferHostSize\_8s32f\_-C4R
  - crosscorrmenormlevel, [1992](#)
- nppiSameNormLevelGetBufferHostSize\_8u32f\_-AC4R
  - crosscorrmenormlevel, [1992](#)
- nppiSameNormLevelGetBufferHostSize\_8u32f\_-C1R
  - crosscorrmenormlevel, [1992](#)
- nppiSameNormLevelGetBufferHostSize\_8u32f\_-C3R
  - crosscorrmenormlevel, [1992](#)
- nppiSameNormLevelGetBufferHostSize\_8u32f\_-C4R
  - crosscorrmenormlevel, [1993](#)
- nppiSameNormLevelGetBufferHostSize\_8u\_-AC4RSfs
  - crosscorrmenormlevel, [1993](#)
- nppiSameNormLevelGetBufferHostSize\_8u\_-C1RSfs
  - crosscorrmenormlevel, [1993](#)
- nppiSameNormLevelGetBufferHostSize\_8u\_-C3RSfs
  - crosscorrmenormlevel, [1994](#)
- nppiSameNormLevelGetBufferHostSize\_8u\_-C4RSfs
  - crosscorrmenormlevel, [1994](#)
- nppiScale\_16s8u\_AC4R
  - image\_scale, [848](#)
- nppiScale\_16s8u\_C1R
  - image\_scale, [848](#)
- nppiScale\_16s8u\_C3R
  - image\_scale, [848](#)
- nppiScale\_16s8u\_C4R
  - image\_scale, [849](#)
- nppiScale\_16u8u\_AC4R
  - image\_scale, [849](#)
- nppiScale\_16u8u\_C1R
  - image\_scale, [849](#)
- nppiScale\_16u8u\_C3R
  - image\_scale, [850](#)
- nppiScale\_16u8u\_C4R
  - image\_scale, [850](#)
- nppiScale\_32f8u\_AC4R
  - image\_scale, [850](#)
- nppiScale\_32f8u\_C1R
  - image\_scale, [851](#)
- nppiScale\_32f8u\_C3R
  - image\_scale, [851](#)
- nppiScale\_32f8u\_C4R
  - image\_scale, [852](#)
- nppiScale\_32s8u\_AC4R
  - image\_scale, [852](#)
- nppiScale\_32s8u\_C1R
  - image\_scale, [852](#)
- nppiScale\_32s8u\_C3R
  - image\_scale, [853](#)
- nppiScale\_32s8u\_C4R
  - image\_scale, [853](#)
- nppiScale\_8u16s\_AC4R
  - image\_scale, [853](#)
- nppiScale\_8u16s\_C1R
  - image\_scale, [854](#)
- nppiScale\_8u16s\_C3R
  - image\_scale, [854](#)
- nppiScale\_8u16s\_C4R
  - image\_scale, [854](#)
- nppiScale\_8u16u\_AC4R
  - image\_scale, [855](#)
- nppiScale\_8u16u\_C1R
  - image\_scale, [855](#)
- nppiScale\_8u16u\_C3R
  - image\_scale, [855](#)
- nppiScale\_8u16u\_C4R
  - image\_scale, [856](#)
- nppiScale\_8u32f\_AC4R
  - image\_scale, [856](#)
- nppiScale\_8u32f\_C1R
  - image\_scale, [856](#)
- nppiScale\_8u32f\_C3R
  - image\_scale, [857](#)
- nppiScale\_8u32f\_C4R
  - image\_scale, [857](#)
- nppiScale\_8u32s\_AC4R
  - image\_scale, [858](#)
- nppiScale\_8u32s\_C1R
  - image\_scale, [858](#)
- nppiScale\_8u32s\_C3R
  - image\_scale, [858](#)
- nppiScale\_8u32s\_C4R
  - image\_scale, [859](#)
- nppiSet\_16s\_AC4MR
  - image\_set, [726](#)
- nppiSet\_16s\_AC4R
  - image\_set, [727](#)
- nppiSet\_16s\_C1MR
  - image\_set, [727](#)
- nppiSet\_16s\_C1R
  - image\_set, [727](#)

- nppiSet\_16s\_C2R
  - image\_set, [728](#)
- nppiSet\_16s\_C3CR
  - image\_set, [728](#)
- nppiSet\_16s\_C3MR
  - image\_set, [728](#)
- nppiSet\_16s\_C3R
  - image\_set, [729](#)
- nppiSet\_16s\_C4CR
  - image\_set, [729](#)
- nppiSet\_16s\_C4MR
  - image\_set, [729](#)
- nppiSet\_16s\_C4R
  - image\_set, [730](#)
- nppiSet\_16sc\_AC4R
  - image\_set, [730](#)
- nppiSet\_16sc\_C1R
  - image\_set, [730](#)
- nppiSet\_16sc\_C2R
  - image\_set, [731](#)
- nppiSet\_16sc\_C3R
  - image\_set, [731](#)
- nppiSet\_16sc\_C4R
  - image\_set, [731](#)
- nppiSet\_16u\_AC4MR
  - image\_set, [732](#)
- nppiSet\_16u\_AC4R
  - image\_set, [732](#)
- nppiSet\_16u\_C1MR
  - image\_set, [732](#)
- nppiSet\_16u\_C1R
  - image\_set, [733](#)
- nppiSet\_16u\_C2R
  - image\_set, [733](#)
- nppiSet\_16u\_C3CR
  - image\_set, [733](#)
- nppiSet\_16u\_C3MR
  - image\_set, [734](#)
- nppiSet\_16u\_C3R
  - image\_set, [734](#)
- nppiSet\_16u\_C4CR
  - image\_set, [734](#)
- nppiSet\_16u\_C4MR
  - image\_set, [735](#)
- nppiSet\_16u\_C4R
  - image\_set, [735](#)
- nppiSet\_32f\_AC4MR
  - image\_set, [735](#)
- nppiSet\_32f\_AC4R
  - image\_set, [736](#)
- nppiSet\_32f\_C1MR
  - image\_set, [736](#)
- nppiSet\_32f\_C1R
  - image\_set, [736](#)
- nppiSet\_32f\_C2R
  - image\_set, [737](#)
- nppiSet\_32f\_C3CR
  - image\_set, [737](#)
- nppiSet\_32f\_C3MR
  - image\_set, [737](#)
- nppiSet\_32f\_C3R
  - image\_set, [738](#)
- nppiSet\_32f\_C4CR
  - image\_set, [738](#)
- nppiSet\_32f\_C4MR
  - image\_set, [738](#)
- nppiSet\_32f\_C4R
  - image\_set, [739](#)
- nppiSet\_32fc\_AC4R
  - image\_set, [739](#)
- nppiSet\_32fc\_C1R
  - image\_set, [739](#)
- nppiSet\_32fc\_C2R
  - image\_set, [740](#)
- nppiSet\_32fc\_C3R
  - image\_set, [740](#)
- nppiSet\_32fc\_C4R
  - image\_set, [740](#)
- nppiSet\_32s\_AC4MR
  - image\_set, [741](#)
- nppiSet\_32s\_AC4R
  - image\_set, [741](#)
- nppiSet\_32s\_C1MR
  - image\_set, [741](#)
- nppiSet\_32s\_C1R
  - image\_set, [742](#)
- nppiSet\_32s\_C2R
  - image\_set, [742](#)
- nppiSet\_32s\_C3CR
  - image\_set, [742](#)
- nppiSet\_32s\_C3MR
  - image\_set, [743](#)
- nppiSet\_32s\_C3R
  - image\_set, [743](#)
- nppiSet\_32s\_C4CR
  - image\_set, [743](#)
- nppiSet\_32s\_C4MR
  - image\_set, [744](#)
- nppiSet\_32s\_C4R
  - image\_set, [744](#)
- nppiSet\_32sc\_AC4R
  - image\_set, [744](#)
- nppiSet\_32sc\_C1R
  - image\_set, [745](#)
- nppiSet\_32sc\_C2R
  - image\_set, [745](#)
- nppiSet\_32sc\_C3R
  - image\_set, [745](#)

nppiSet\_32sc\_C4R  
     image\_set, [746](#)  
 nppiSet\_32u\_AC4R  
     image\_set, [746](#)  
 nppiSet\_32u\_C1R  
     image\_set, [746](#)  
 nppiSet\_32u\_C2R  
     image\_set, [747](#)  
 nppiSet\_32u\_C3R  
     image\_set, [747](#)  
 nppiSet\_32u\_C4R  
     image\_set, [747](#)  
 nppiSet\_8s\_AC4R  
     image\_set, [748](#)  
 nppiSet\_8s\_C1R  
     image\_set, [748](#)  
 nppiSet\_8s\_C2R  
     image\_set, [748](#)  
 nppiSet\_8s\_C3R  
     image\_set, [749](#)  
 nppiSet\_8s\_C4R  
     image\_set, [749](#)  
 nppiSet\_8u\_AC4MR  
     image\_set, [749](#)  
 nppiSet\_8u\_AC4R  
     image\_set, [750](#)  
 nppiSet\_8u\_C1MR  
     image\_set, [750](#)  
 nppiSet\_8u\_C1R  
     image\_set, [750](#)  
 nppiSet\_8u\_C2R  
     image\_set, [751](#)  
 nppiSet\_8u\_C3CR  
     image\_set, [751](#)  
 nppiSet\_8u\_C3MR  
     image\_set, [751](#)  
 nppiSet\_8u\_C3R  
     image\_set, [752](#)  
 nppiSet\_8u\_C4CR  
     image\_set, [752](#)  
 nppiSet\_8u\_C4MR  
     image\_set, [752](#)  
 nppiSet\_8u\_C4R  
     image\_set, [753](#)  
 NppiSize, [2633](#)  
     height, [2633](#)  
     width, [2633](#)  
 nppiSqr\_16s\_AC4IRSfs  
     image\_sqr, [333](#)  
 nppiSqr\_16s\_AC4RSfs  
     image\_sqr, [333](#)  
 nppiSqr\_16s\_C1IRSfs  
     image\_sqr, [333](#)  
 nppiSqr\_16s\_C1RSfs  
     image\_sqr, [333](#)  
 nppiSqr\_16s\_C4IRSfs  
     image\_sqr, [333](#)  
 nppiSqr\_16s\_C4RSfs  
     image\_sqr, [335](#)  
 nppiSqr\_16u\_AC4IRSfs  
     image\_sqr, [335](#)  
 nppiSqr\_16u\_AC4RSfs  
     image\_sqr, [335](#)  
 nppiSqr\_16u\_C1IRSfs  
     image\_sqr, [336](#)  
 nppiSqr\_16u\_C1RSfs  
     image\_sqr, [336](#)  
 nppiSqr\_16u\_C3IRSfs  
     image\_sqr, [337](#)  
 nppiSqr\_16u\_C3RSfs  
     image\_sqr, [337](#)  
 nppiSqr\_16u\_C4IRSfs  
     image\_sqr, [337](#)  
 nppiSqr\_16u\_C4RSfs  
     image\_sqr, [338](#)  
 nppiSqr\_32f\_AC4IR  
     image\_sqr, [338](#)  
 nppiSqr\_32f\_AC4R  
     image\_sqr, [338](#)  
 nppiSqr\_32f\_C1IR  
     image\_sqr, [339](#)  
 nppiSqr\_32f\_C1R  
     image\_sqr, [339](#)  
 nppiSqr\_32f\_C3IR  
     image\_sqr, [339](#)  
 nppiSqr\_32f\_C3R  
     image\_sqr, [339](#)  
 nppiSqr\_32f\_C4IR  
     image\_sqr, [340](#)  
 nppiSqr\_32f\_C4R  
     image\_sqr, [340](#)  
 nppiSqr\_8u\_AC4IRSfs  
     image\_sqr, [340](#)  
 nppiSqr\_8u\_AC4RSfs  
     image\_sqr, [341](#)  
 nppiSqr\_8u\_C1IRSfs  
     image\_sqr, [341](#)  
 nppiSqr\_8u\_C1RSfs  
     image\_sqr, [341](#)  
 nppiSqr\_8u\_C3IRSfs  
     image\_sqr, [342](#)  
 nppiSqr\_8u\_C3RSfs  
     image\_sqr, [342](#)  
 nppiSqr\_8u\_C4IRSfs



- image\_sqr, [342](#)
- nppiSqr\_8u\_C4RSfs
  - image\_sqr, [343](#)
- nppiSqrDistanceFull\_Norm\_16u32f\_AC4R
  - sqrdistancefullnorm, [1888](#)
- nppiSqrDistanceFull\_Norm\_16u32f\_C1R
  - sqrdistancefullnorm, [1888](#)
- nppiSqrDistanceFull\_Norm\_16u32f\_C3R
  - sqrdistancefullnorm, [1888](#)
- nppiSqrDistanceFull\_Norm\_16u32f\_C4R
  - sqrdistancefullnorm, [1889](#)
- nppiSqrDistanceFull\_Norm\_32f\_AC4R
  - sqrdistancefullnorm, [1889](#)
- nppiSqrDistanceFull\_Norm\_32f\_C1R
  - sqrdistancefullnorm, [1890](#)
- nppiSqrDistanceFull\_Norm\_32f\_C3R
  - sqrdistancefullnorm, [1890](#)
- nppiSqrDistanceFull\_Norm\_32f\_C4R
  - sqrdistancefullnorm, [1891](#)
- nppiSqrDistanceFull\_Norm\_8s32f\_AC4R
  - sqrdistancefullnorm, [1891](#)
- nppiSqrDistanceFull\_Norm\_8s32f\_C1R
  - sqrdistancefullnorm, [1891](#)
- nppiSqrDistanceFull\_Norm\_8s32f\_C3R
  - sqrdistancefullnorm, [1892](#)
- nppiSqrDistanceFull\_Norm\_8s32f\_C4R
  - sqrdistancefullnorm, [1892](#)
- nppiSqrDistanceFull\_Norm\_8u32f\_AC4R
  - sqrdistancefullnorm, [1893](#)
- nppiSqrDistanceFull\_Norm\_8u32f\_C1R
  - sqrdistancefullnorm, [1893](#)
- nppiSqrDistanceFull\_Norm\_8u32f\_C3R
  - sqrdistancefullnorm, [1894](#)
- nppiSqrDistanceFull\_Norm\_8u32f\_C4R
  - sqrdistancefullnorm, [1894](#)
- nppiSqrDistanceFull\_Norm\_8u\_AC4RSfs
  - sqrdistancefullnorm, [1894](#)
- nppiSqrDistanceFull\_Norm\_8u\_C1RSfs
  - sqrdistancefullnorm, [1895](#)
- nppiSqrDistanceFull\_Norm\_8u\_C3RSfs
  - sqrdistancefullnorm, [1895](#)
- nppiSqrDistanceFull\_Norm\_8u\_C4RSfs
  - sqrdistancefullnorm, [1896](#)
- nppiSqrDistanceSame\_Norm\_16u32f\_AC4R
  - sqrdistancesamenorm, [1899](#)
- nppiSqrDistanceSame\_Norm\_16u32f\_C1R
  - sqrdistancesamenorm, [1899](#)
- nppiSqrDistanceSame\_Norm\_16u32f\_C3R
  - sqrdistancesamenorm, [1900](#)
- nppiSqrDistanceSame\_Norm\_16u32f\_C4R
  - sqrdistancesamenorm, [1900](#)
- nppiSqrDistanceSame\_Norm\_32f\_AC4R
  - sqrdistancesamenorm, [1900](#)
- nppiSqrDistanceSame\_Norm\_32f\_C1R
  - sqrdistancesamenorm, [1901](#)
- nppiSqrDistanceSame\_Norm\_32f\_C3R
  - sqrdistancesamenorm, [1901](#)
- nppiSqrDistanceSame\_Norm\_32f\_C4R
  - sqrdistancesamenorm, [1902](#)
- nppiSqrDistanceSame\_Norm\_8s32f\_AC4R
  - sqrdistancesamenorm, [1902](#)
- nppiSqrDistanceSame\_Norm\_8s32f\_C1R
  - sqrdistancesamenorm, [1903](#)
- nppiSqrDistanceSame\_Norm\_8s32f\_C3R
  - sqrdistancesamenorm, [1903](#)
- nppiSqrDistanceSame\_Norm\_8s32f\_C4R
  - sqrdistancesamenorm, [1903](#)
- nppiSqrDistanceSame\_Norm\_8u32f\_AC4R
  - sqrdistancesamenorm, [1904](#)
- nppiSqrDistanceSame\_Norm\_8u32f\_C1R
  - sqrdistancesamenorm, [1904](#)
- nppiSqrDistanceSame\_Norm\_8u32f\_C3R
  - sqrdistancesamenorm, [1905](#)
- nppiSqrDistanceSame\_Norm\_8u32f\_C4R
  - sqrdistancesamenorm, [1905](#)
- nppiSqrDistanceSame\_Norm\_8u\_AC4RSfs
  - sqrdistancesamenorm, [1906](#)
- nppiSqrDistanceSame\_Norm\_8u\_C1RSfs
  - sqrdistancesamenorm, [1906](#)
- nppiSqrDistanceSame\_Norm\_8u\_C3RSfs
  - sqrdistancesamenorm, [1907](#)
- nppiSqrDistanceSame\_Norm\_8u\_C4RSfs
  - sqrdistancesamenorm, [1907](#)
- nppiSqrDistanceValid\_Norm\_16u32f\_AC4R
  - sqrdistancevalidnorm, [1910](#)
- nppiSqrDistanceValid\_Norm\_16u32f\_C1R
  - sqrdistancevalidnorm, [1910](#)
- nppiSqrDistanceValid\_Norm\_16u32f\_C3R
  - sqrdistancevalidnorm, [1911](#)
- nppiSqrDistanceValid\_Norm\_16u32f\_C4R
  - sqrdistancevalidnorm, [1911](#)
- nppiSqrDistanceValid\_Norm\_32f\_AC4R
  - sqrdistancevalidnorm, [1911](#)
- nppiSqrDistanceValid\_Norm\_32f\_C1R
  - sqrdistancevalidnorm, [1912](#)
- nppiSqrDistanceValid\_Norm\_32f\_C3R
  - sqrdistancevalidnorm, [1912](#)
- nppiSqrDistanceValid\_Norm\_32f\_C4R
  - sqrdistancevalidnorm, [1913](#)
- nppiSqrDistanceValid\_Norm\_8s32f\_AC4R
  - sqrdistancevalidnorm, [1913](#)
- nppiSqrDistanceValid\_Norm\_8s32f\_C1R
  - sqrdistancevalidnorm, [1914](#)
- nppiSqrDistanceValid\_Norm\_8s32f\_C3R
  - sqrdistancevalidnorm, [1914](#)
- nppiSqrDistanceValid\_Norm\_8s32f\_C4R
  - sqrdistancevalidnorm, [1914](#)
- nppiSqrDistanceValid\_Norm\_8u32f\_AC4R

sqrdistancevalidnorm, [1915](#)  
 nppiSqrDistanceValid\_Norm\_8u32f\_C1R  
     sqrdistancevalidnorm, [1915](#)  
 nppiSqrDistanceValid\_Norm\_8u32f\_C3R  
     sqrdistancevalidnorm, [1916](#)  
 nppiSqrDistanceValid\_Norm\_8u32f\_C4R  
     sqrdistancevalidnorm, [1916](#)  
 nppiSqrDistanceValid\_Norm\_8u\_AC4RSfs  
     sqrdistancevalidnorm, [1917](#)  
 nppiSqrDistanceValid\_Norm\_8u\_C1RSfs  
     sqrdistancevalidnorm, [1917](#)  
 nppiSqrDistanceValid\_Norm\_8u\_C3RSfs  
     sqrdistancevalidnorm, [1918](#)  
 nppiSqrDistanceValid\_Norm\_8u\_C4RSfs  
     sqrdistancevalidnorm, [1918](#)  
 nppiSqrIntegral\_8u32f64f\_C1R  
     image\_sqrtintegral, [1848](#)  
 nppiSqrIntegral\_8u32s64f\_C1R  
     image\_sqrtintegral, [1849](#)  
 nppiSqrIntegral\_8u32s\_C1R  
     image\_sqrtintegral, [1849](#)  
 nppiSqrt\_16s\_AC4IRSfs  
     image\_sqrt, [346](#)  
 nppiSqrt\_16s\_AC4RSfs  
     image\_sqrt, [346](#)  
 nppiSqrt\_16s\_C1IRSfs  
     image\_sqrt, [347](#)  
 nppiSqrt\_16s\_C1RSfs  
     image\_sqrt, [347](#)  
 nppiSqrt\_16s\_C3IRSfs  
     image\_sqrt, [348](#)  
 nppiSqrt\_16s\_C3RSfs  
     image\_sqrt, [348](#)  
 nppiSqrt\_16u\_AC4IRSfs  
     image\_sqrt, [348](#)  
 nppiSqrt\_16u\_AC4RSfs  
     image\_sqrt, [349](#)  
 nppiSqrt\_16u\_C1IRSfs  
     image\_sqrt, [349](#)  
 nppiSqrt\_16u\_C1RSfs  
     image\_sqrt, [349](#)  
 nppiSqrt\_16u\_C3IRSfs  
     image\_sqrt, [350](#)  
 nppiSqrt\_16u\_C3RSfs  
     image\_sqrt, [350](#)  
 nppiSqrt\_32f\_AC4IR  
     image\_sqrt, [350](#)  
 nppiSqrt\_32f\_AC4R  
     image\_sqrt, [351](#)  
 nppiSqrt\_32f\_C1IR  
     image\_sqrt, [351](#)  
 nppiSqrt\_32f\_C1R  
     image\_sqrt, [351](#)  
 nppiSqrt\_32f\_C3IR  
     image\_sqrt, [352](#)  
 nppiSqrt\_32f\_C3R  
     image\_sqrt, [352](#)  
 nppiSqrt\_32f\_C4IR  
     image\_sqrt, [352](#)  
 nppiSqrt\_32f\_C4R  
     image\_sqrt, [353](#)  
 nppiSqrt\_8u\_AC4IRSfs  
     image\_sqrt, [353](#)  
 nppiSqrt\_8u\_AC4RSfs  
     image\_sqrt, [353](#)  
 nppiSqrt\_8u\_C1IRSfs  
     image\_sqrt, [354](#)  
 nppiSqrt\_8u\_C1RSfs  
     image\_sqrt, [354](#)  
 nppiSqrt\_8u\_C3IRSfs  
     image\_sqrt, [355](#)  
 nppiSqrt\_8u\_C3RSfs  
     image\_sqrt, [355](#)  
 nppiSub\_16s\_AC4IRSfs  
     image\_sub, [251](#)  
 nppiSub\_16s\_AC4RSfs  
     image\_sub, [252](#)  
 nppiSub\_16s\_C1IRSfs  
     image\_sub, [252](#)  
 nppiSub\_16s\_C1RSfs  
     image\_sub, [252](#)  
 nppiSub\_16s\_C3IRSfs  
     image\_sub, [253](#)  
 nppiSub\_16s\_C3RSfs  
     image\_sub, [253](#)  
 nppiSub\_16s\_C4IRSfs  
     image\_sub, [254](#)  
 nppiSub\_16s\_C4RSfs  
     image\_sub, [254](#)  
 nppiSub\_16sc\_AC4IRSfs  
     image\_sub, [254](#)  
 nppiSub\_16sc\_AC4RSfs  
     image\_sub, [255](#)  
 nppiSub\_16sc\_C1IRSfs  
     image\_sub, [255](#)  
 nppiSub\_16sc\_C1RSfs  
     image\_sub, [256](#)  
 nppiSub\_16sc\_C3IRSfs  
     image\_sub, [256](#)  
 nppiSub\_16sc\_C3RSfs  
     image\_sub, [256](#)  
 nppiSub\_16u\_AC4IRSfs  
     image\_sub, [257](#)  
 nppiSub\_16u\_AC4RSfs  
     image\_sub, [257](#)  
 nppiSub\_16u\_C1IRSfs  
     image\_sub, [258](#)  
 nppiSub\_16u\_C1RSfs



- image\_sub, [258](#)
- nppiSub\_16u\_C3IRSfs
  - image\_sub, [259](#)
- nppiSub\_16u\_C3RSfs
  - image\_sub, [259](#)
- nppiSub\_16u\_C4IRSfs
  - image\_sub, [259](#)
- nppiSub\_16u\_C4RSfs
  - image\_sub, [260](#)
- nppiSub\_32f\_AC4IR
  - image\_sub, [260](#)
- nppiSub\_32f\_AC4R
  - image\_sub, [261](#)
- nppiSub\_32f\_C1IR
  - image\_sub, [261](#)
- nppiSub\_32f\_C1R
  - image\_sub, [261](#)
- nppiSub\_32f\_C3IR
  - image\_sub, [262](#)
- nppiSub\_32f\_C3R
  - image\_sub, [262](#)
- nppiSub\_32f\_C4IR
  - image\_sub, [263](#)
- nppiSub\_32f\_C4R
  - image\_sub, [263](#)
- nppiSub\_32fc\_AC4IR
  - image\_sub, [263](#)
- nppiSub\_32fc\_AC4R
  - image\_sub, [264](#)
- nppiSub\_32fc\_C1IR
  - image\_sub, [264](#)
- nppiSub\_32fc\_C1R
  - image\_sub, [265](#)
- nppiSub\_32fc\_C3IR
  - image\_sub, [265](#)
- nppiSub\_32fc\_C3R
  - image\_sub, [265](#)
- nppiSub\_32fc\_C4IR
  - image\_sub, [266](#)
- nppiSub\_32fc\_C4R
  - image\_sub, [266](#)
- nppiSub\_32s\_C1IRSfs
  - image\_sub, [267](#)
- nppiSub\_32s\_C1R
  - image\_sub, [267](#)
- nppiSub\_32s\_C1RSfs
  - image\_sub, [267](#)
- nppiSub\_32s\_C3IRSfs
  - image\_sub, [268](#)
- nppiSub\_32s\_C3RSfs
  - image\_sub, [268](#)
- nppiSub\_32s\_C4IRSfs
  - image\_sub, [269](#)
- nppiSub\_32s\_C4RSfs
  - image\_sub, [269](#)
- nppiSub\_32sc\_AC4IRSfs
  - image\_sub, [270](#)
- nppiSub\_32sc\_AC4RSfs
  - image\_sub, [270](#)
- nppiSub\_32sc\_C1IRSfs
  - image\_sub, [270](#)
- nppiSub\_32sc\_C1RSfs
  - image\_sub, [271](#)
- nppiSub\_32sc\_C3IRSfs
  - image\_sub, [271](#)
- nppiSub\_32sc\_C3RSfs
  - image\_sub, [272](#)
- nppiSub\_8u\_AC4IRSfs
  - image\_sub, [272](#)
- nppiSub\_8u\_AC4RSfs
  - image\_sub, [272](#)
- nppiSub\_8u\_C1IRSfs
  - image\_sub, [273](#)
- nppiSub\_8u\_C1RSfs
  - image\_sub, [273](#)
- nppiSub\_8u\_C3IRSfs
  - image\_sub, [274](#)
- nppiSub\_8u\_C3RSfs
  - image\_sub, [274](#)
- nppiSub\_8u\_C4IRSfs
  - image\_sub, [274](#)
- nppiSub\_8u\_C4RSfs
  - image\_sub, [275](#)
- nppiSubC\_16s\_AC4IRSfs
  - image\_subc, [119](#)
- nppiSubC\_16s\_AC4RSfs
  - image\_subc, [119](#)
- nppiSubC\_16s\_C1IRSfs
  - image\_subc, [119](#)
- nppiSubC\_16s\_C1RSfs
  - image\_subc, [120](#)
- nppiSubC\_16s\_C3IRSfs
  - image\_subc, [120](#)
- nppiSubC\_16s\_C3RSfs
  - image\_subc, [120](#)
- nppiSubC\_16s\_C4IRSfs
  - image\_subc, [121](#)
- nppiSubC\_16s\_C4RSfs
  - image\_subc, [121](#)
- nppiSubC\_16sc\_AC4IRSfs
  - image\_subc, [122](#)
- nppiSubC\_16sc\_AC4RSfs
  - image\_subc, [122](#)
- nppiSubC\_16sc\_C1IRSfs
  - image\_subc, [122](#)
- nppiSubC\_16sc\_C1RSfs
  - image\_subc, [123](#)
- nppiSubC\_16sc\_C3IRSfs

image\_subc, [123](#)  
 nppiSubC\_16sc\_C3RSfs  
   image\_subc, [124](#)  
 nppiSubC\_16u\_AC4IRSfs  
   image\_subc, [124](#)  
 nppiSubC\_16u\_AC4RSfs  
   image\_subc, [124](#)  
 nppiSubC\_16u\_C1IRSfs  
   image\_subc, [125](#)  
 nppiSubC\_16u\_C1RSfs  
   image\_subc, [125](#)  
 nppiSubC\_16u\_C3IRSfs  
   image\_subc, [126](#)  
 nppiSubC\_16u\_C3RSfs  
   image\_subc, [126](#)  
 nppiSubC\_16u\_C4IRSfs  
   image\_subc, [126](#)  
 nppiSubC\_16u\_C4RSfs  
   image\_subc, [127](#)  
 nppiSubC\_32f\_AC4IR  
   image\_subc, [127](#)  
 nppiSubC\_32f\_AC4R  
   image\_subc, [127](#)  
 nppiSubC\_32f\_C1IR  
   image\_subc, [128](#)  
 nppiSubC\_32f\_C1R  
   image\_subc, [128](#)  
 nppiSubC\_32f\_C3IR  
   image\_subc, [128](#)  
 nppiSubC\_32f\_C3R  
   image\_subc, [129](#)  
 nppiSubC\_32f\_C4IR  
   image\_subc, [129](#)  
 nppiSubC\_32f\_C4R  
   image\_subc, [129](#)  
 nppiSubC\_32fc\_AC4IR  
   image\_subc, [130](#)  
 nppiSubC\_32fc\_AC4R  
   image\_subc, [130](#)  
 nppiSubC\_32fc\_C1IR  
   image\_subc, [130](#)  
 nppiSubC\_32fc\_C1R  
   image\_subc, [131](#)  
 nppiSubC\_32fc\_C3IR  
   image\_subc, [131](#)  
 nppiSubC\_32fc\_C3R  
   image\_subc, [131](#)  
 nppiSubC\_32fc\_C4IR  
   image\_subc, [132](#)  
 nppiSubC\_32fc\_C4R  
   image\_subc, [132](#)  
 nppiSubC\_32s\_C1IRSfs  
   image\_subc, [133](#)  
 nppiSubC\_32s\_C1RSfs

image\_subc, [133](#)  
 nppiSubC\_32s\_C3IRSfs  
   image\_subc, [133](#)  
 nppiSubC\_32s\_C3RSfs  
   image\_subc, [134](#)  
 nppiSubC\_32sc\_AC4IRSfs  
   image\_subc, [134](#)  
 nppiSubC\_32sc\_AC4RSfs  
   image\_subc, [134](#)  
 nppiSubC\_32sc\_C1IRSfs  
   image\_subc, [135](#)  
 nppiSubC\_32sc\_C1RSfs  
   image\_subc, [135](#)  
 nppiSubC\_32sc\_C3IRSfs  
   image\_subc, [136](#)  
 nppiSubC\_32sc\_C3RSfs  
   image\_subc, [136](#)  
 nppiSubC\_8u\_AC4IRSfs  
   image\_subc, [136](#)  
 nppiSubC\_8u\_AC4RSfs  
   image\_subc, [137](#)  
 nppiSubC\_8u\_C1IRSfs  
   image\_subc, [137](#)  
 nppiSubC\_8u\_C1RSfs  
   image\_subc, [138](#)  
 nppiSubC\_8u\_C3IRSfs  
   image\_subc, [138](#)  
 nppiSubC\_8u\_C3RSfs  
   image\_subc, [138](#)  
 nppiSubC\_8u\_C4IRSfs  
   image\_subc, [139](#)  
 nppiSubC\_8u\_C4RSfs  
   image\_subc, [139](#)  
 nppiSum\_16s\_AC4R  
   image\_sum, [1463](#)  
 nppiSum\_16s\_C1R  
   image\_sum, [1463](#)  
 nppiSum\_16s\_C3R  
   image\_sum, [1463](#)  
 nppiSum\_16s\_C4R  
   image\_sum, [1464](#)  
 nppiSum\_16u\_AC4R  
   image\_sum, [1464](#)  
 nppiSum\_16u\_C1R  
   image\_sum, [1464](#)  
 nppiSum\_16u\_C3R  
   image\_sum, [1465](#)  
 nppiSum\_16u\_C4R  
   image\_sum, [1465](#)  
 nppiSum\_32f\_AC4R  
   image\_sum, [1465](#)  
 nppiSum\_32f\_C1R  
   image\_sum, [1466](#)  
 nppiSum\_32f\_C3R

- image\_sum, [1466](#)
- nppiSum\_32f\_C4R
  - image\_sum, [1466](#)
- nppiSum\_8u64s\_C1R
  - image\_sum, [1467](#)
- nppiSum\_8u64s\_C4R
  - image\_sum, [1467](#)
- nppiSum\_8u\_AC4R
  - image\_sum, [1468](#)
- nppiSum\_8u\_C1R
  - image\_sum, [1468](#)
- nppiSum\_8u\_C3R
  - image\_sum, [1468](#)
- nppiSum\_8u\_C4R
  - image\_sum, [1469](#)
- nppiSumGetBufferHostSize\_16s\_AC4R
  - image\_sum, [1469](#)
- nppiSumGetBufferHostSize\_16s\_C1R
  - image\_sum, [1469](#)
- nppiSumGetBufferHostSize\_16s\_C3R
  - image\_sum, [1470](#)
- nppiSumGetBufferHostSize\_16s\_C4R
  - image\_sum, [1470](#)
- nppiSumGetBufferHostSize\_16u\_AC4R
  - image\_sum, [1470](#)
- nppiSumGetBufferHostSize\_16u\_C1R
  - image\_sum, [1471](#)
- nppiSumGetBufferHostSize\_16u\_C3R
  - image\_sum, [1471](#)
- nppiSumGetBufferHostSize\_16u\_C4R
  - image\_sum, [1471](#)
- nppiSumGetBufferHostSize\_32f\_AC4R
  - image\_sum, [1471](#)
- nppiSumGetBufferHostSize\_32f\_C1R
  - image\_sum, [1472](#)
- nppiSumGetBufferHostSize\_32f\_C3R
  - image\_sum, [1472](#)
- nppiSumGetBufferHostSize\_32f\_C4R
  - image\_sum, [1472](#)
- nppiSumGetBufferHostSize\_8u64s\_C1R
  - image\_sum, [1473](#)
- nppiSumGetBufferHostSize\_8u64s\_C4R
  - image\_sum, [1473](#)
- nppiSumGetBufferHostSize\_8u\_AC4R
  - image\_sum, [1473](#)
- nppiSumGetBufferHostSize\_8u\_C1R
  - image\_sum, [1473](#)
- nppiSumGetBufferHostSize\_8u\_C3R
  - image\_sum, [1474](#)
- nppiSumGetBufferHostSize\_8u\_C4R
  - image\_sum, [1474](#)
- nppiSumWindowColumn\_16s32f\_C1R
  - image\_1D\_window\_sum, [1047](#)
- nppiSumWindowColumn\_16s32f\_C3R
  - image\_1D\_window\_sum, [1048](#)
- nppiSumWindowColumn\_16s32f\_C4R
  - image\_1D\_window\_sum, [1048](#)
- nppiSumWindowColumn\_16u32f\_C1R
  - image\_1D\_window\_sum, [1049](#)
- nppiSumWindowColumn\_16u32f\_C3R
  - image\_1D\_window\_sum, [1049](#)
- nppiSumWindowColumn\_16u32f\_C4R
  - image\_1D\_window\_sum, [1050](#)
- nppiSumWindowColumn\_8u32f\_C1R
  - image\_1D\_window\_sum, [1050](#)
- nppiSumWindowColumn\_8u32f\_C3R
  - image\_1D\_window\_sum, [1050](#)
- nppiSumWindowColumn\_8u32f\_C4R
  - image\_1D\_window\_sum, [1051](#)
- nppiSumWindowRow\_16s32f\_C1R
  - image\_1D\_window\_sum, [1051](#)
- nppiSumWindowRow\_16s32f\_C3R
  - image\_1D\_window\_sum, [1052](#)
- nppiSumWindowRow\_16s32f\_C4R
  - image\_1D\_window\_sum, [1052](#)
- nppiSumWindowRow\_16u32f\_C1R
  - image\_1D\_window\_sum, [1053](#)
- nppiSumWindowRow\_16u32f\_C3R
  - image\_1D\_window\_sum, [1053](#)
- nppiSumWindowRow\_16u32f\_C4R
  - image\_1D\_window\_sum, [1054](#)
- nppiSumWindowRow\_8u32f\_C1R
  - image\_1D\_window\_sum, [1054](#)
- nppiSumWindowRow\_8u32f\_C3R
  - image\_1D\_window\_sum, [1055](#)
- nppiSumWindowRow\_8u32f\_C4R
  - image\_1D\_window\_sum, [1055](#)
- nppiSwapChannels\_16s\_AC4R
  - image\_swap\_channels, [926](#)
- nppiSwapChannels\_16s\_C3C4R
  - image\_swap\_channels, [926](#)
- nppiSwapChannels\_16s\_C3IR
  - image\_swap\_channels, [926](#)
- nppiSwapChannels\_16s\_C3R
  - image\_swap\_channels, [927](#)
- nppiSwapChannels\_16s\_C4C3R
  - image\_swap\_channels, [927](#)
- nppiSwapChannels\_16s\_C4IR
  - image\_swap\_channels, [928](#)
- nppiSwapChannels\_16s\_C4R
  - image\_swap\_channels, [928](#)
- nppiSwapChannels\_16u\_AC4R
  - image\_swap\_channels, [928](#)
- nppiSwapChannels\_16u\_C3C4R
  - image\_swap\_channels, [929](#)
- nppiSwapChannels\_16u\_C3IR
  - image\_swap\_channels, [929](#)
- nppiSwapChannels\_16u\_C3R
  - image\_swap\_channels, [929](#)

- image\_swap\_channels, [930](#)
- nppiSwapChannels\_16u\_C4C3R
  - image\_swap\_channels, [930](#)
- nppiSwapChannels\_16u\_C4IR
  - image\_swap\_channels, [931](#)
- nppiSwapChannels\_16u\_C4R
  - image\_swap\_channels, [931](#)
- nppiSwapChannels\_32f\_AC4R
  - image\_swap\_channels, [931](#)
- nppiSwapChannels\_32f\_C3C4R
  - image\_swap\_channels, [932](#)
- nppiSwapChannels\_32f\_C3IR
  - image\_swap\_channels, [932](#)
- nppiSwapChannels\_32f\_C3R
  - image\_swap\_channels, [933](#)
- nppiSwapChannels\_32f\_C4C3R
  - image\_swap\_channels, [933](#)
- nppiSwapChannels\_32f\_C4IR
  - image\_swap\_channels, [934](#)
- nppiSwapChannels\_32f\_C4R
  - image\_swap\_channels, [934](#)
- nppiSwapChannels\_32s\_AC4R
  - image\_swap\_channels, [934](#)
- nppiSwapChannels\_32s\_C3C4R
  - image\_swap\_channels, [935](#)
- nppiSwapChannels\_32s\_C3IR
  - image\_swap\_channels, [935](#)
- nppiSwapChannels\_32s\_C3R
  - image\_swap\_channels, [936](#)
- nppiSwapChannels\_32s\_C4C3R
  - image\_swap\_channels, [936](#)
- nppiSwapChannels\_32s\_C4IR
  - image\_swap\_channels, [937](#)
- nppiSwapChannels\_32s\_C4R
  - image\_swap\_channels, [937](#)
- nppiSwapChannels\_8u\_AC4R
  - image\_swap\_channels, [937](#)
- nppiSwapChannels\_8u\_C3C4R
  - image\_swap\_channels, [938](#)
- nppiSwapChannels\_8u\_C3IR
  - image\_swap\_channels, [938](#)
- nppiSwapChannels\_8u\_C3R
  - image\_swap\_channels, [939](#)
- nppiSwapChannels\_8u\_C4C3R
  - image\_swap\_channels, [939](#)
- nppiSwapChannels\_8u\_C4IR
  - image\_swap\_channels, [940](#)
- nppiSwapChannels\_8u\_C4R
  - image\_swap\_channels, [940](#)
- nppiThreshold\_16s\_AC4IR
  - image\_threshold\_operations, [2145](#)
- nppiThreshold\_16s\_AC4R
  - image\_threshold\_operations, [2145](#)
- nppiThreshold\_16s\_C1IR
  - image\_threshold\_operations, [2146](#)
- nppiThreshold\_16s\_C1R
  - image\_threshold\_operations, [2146](#)
- nppiThreshold\_16s\_C3IR
  - image\_threshold\_operations, [2147](#)
- nppiThreshold\_16s\_C3R
  - image\_threshold\_operations, [2147](#)
- nppiThreshold\_16u\_AC4IR
  - image\_threshold\_operations, [2148](#)
- nppiThreshold\_16u\_AC4R
  - image\_threshold\_operations, [2148](#)
- nppiThreshold\_16u\_C1IR
  - image\_threshold\_operations, [2148](#)
- nppiThreshold\_16u\_C1R
  - image\_threshold\_operations, [2149](#)
- nppiThreshold\_16u\_C3IR
  - image\_threshold\_operations, [2149](#)
- nppiThreshold\_16u\_C3R
  - image\_threshold\_operations, [2150](#)
- nppiThreshold\_32f\_AC4IR
  - image\_threshold\_operations, [2150](#)
- nppiThreshold\_32f\_AC4R
  - image\_threshold\_operations, [2151](#)
- nppiThreshold\_32f\_C1IR
  - image\_threshold\_operations, [2151](#)
- nppiThreshold\_32f\_C1R
  - image\_threshold\_operations, [2152](#)
- nppiThreshold\_32f\_C3IR
  - image\_threshold\_operations, [2152](#)
- nppiThreshold\_32f\_C3R
  - image\_threshold\_operations, [2152](#)
- nppiThreshold\_8u\_AC4IR
  - image\_threshold\_operations, [2153](#)
- nppiThreshold\_8u\_AC4R
  - image\_threshold\_operations, [2153](#)
- nppiThreshold\_8u\_C1IR
  - image\_threshold\_operations, [2154](#)
- nppiThreshold\_8u\_C1R
  - image\_threshold\_operations, [2154](#)
- nppiThreshold\_8u\_C3IR
  - image\_threshold\_operations, [2155](#)
- nppiThreshold\_8u\_C3R
  - image\_threshold\_operations, [2155](#)
- nppiThreshold\_GT\_16s\_AC4IR
  - image\_threshold\_operations, [2156](#)
- nppiThreshold\_GT\_16s\_AC4R
  - image\_threshold\_operations, [2156](#)
- nppiThreshold\_GT\_16s\_C1IR
  - image\_threshold\_operations, [2157](#)
- nppiThreshold\_GT\_16s\_C1R
  - image\_threshold\_operations, [2157](#)
- nppiThreshold\_GT\_16s\_C3IR
  - image\_threshold\_operations, [2157](#)
- nppiThreshold\_GT\_16s\_C3R
  - image\_threshold\_operations, [2157](#)

- image\_threshold\_operations, 2158
- npptThreshold\_GT\_16u\_AC4IR
  - image\_threshold\_operations, 2158
- npptThreshold\_GT\_16u\_AC4R
  - image\_threshold\_operations, 2159
- npptThreshold\_GT\_16u\_C1IR
  - image\_threshold\_operations, 2159
- npptThreshold\_GT\_16u\_C1R
  - image\_threshold\_operations, 2159
- npptThreshold\_GT\_16u\_C3IR
  - image\_threshold\_operations, 2160
- npptThreshold\_GT\_16u\_C3R
  - image\_threshold\_operations, 2160
- npptThreshold\_GT\_32f\_AC4IR
  - image\_threshold\_operations, 2161
- npptThreshold\_GT\_32f\_AC4R
  - image\_threshold\_operations, 2161
- npptThreshold\_GT\_32f\_C1IR
  - image\_threshold\_operations, 2161
- npptThreshold\_GT\_32f\_C1R
  - image\_threshold\_operations, 2162
- npptThreshold\_GT\_32f\_C3IR
  - image\_threshold\_operations, 2162
- npptThreshold\_GT\_32f\_C3R
  - image\_threshold\_operations, 2163
- npptThreshold\_GT\_8u\_AC4IR
  - image\_threshold\_operations, 2163
- npptThreshold\_GT\_8u\_AC4R
  - image\_threshold\_operations, 2163
- npptThreshold\_GT\_8u\_C1IR
  - image\_threshold\_operations, 2164
- npptThreshold\_GT\_8u\_C1R
  - image\_threshold\_operations, 2164
- npptThreshold\_GT\_8u\_C3IR
  - image\_threshold\_operations, 2165
- npptThreshold\_GT\_8u\_C3R
  - image\_threshold\_operations, 2165
- npptThreshold\_GTVal\_16s\_AC4IR
  - image\_threshold\_operations, 2165
- npptThreshold\_GTVal\_16s\_AC4R
  - image\_threshold\_operations, 2166
- npptThreshold\_GTVal\_16s\_C1IR
  - image\_threshold\_operations, 2166
- npptThreshold\_GTVal\_16s\_C1R
  - image\_threshold\_operations, 2167
- npptThreshold\_GTVal\_16s\_C3IR
  - image\_threshold\_operations, 2167
- npptThreshold\_GTVal\_16s\_C3R
  - image\_threshold\_operations, 2167
- npptThreshold\_GTVal\_16u\_AC4IR
  - image\_threshold\_operations, 2168
- npptThreshold\_GTVal\_16u\_AC4R
  - image\_threshold\_operations, 2168
- npptThreshold\_GTVal\_16u\_C1IR
  - image\_threshold\_operations, 2169
- npptThreshold\_GTVal\_16u\_C1R
  - image\_threshold\_operations, 2169
- npptThreshold\_GTVal\_16u\_C3IR
  - image\_threshold\_operations, 2170
- npptThreshold\_GTVal\_16u\_C3R
  - image\_threshold\_operations, 2170
- npptThreshold\_GTVal\_32f\_AC4IR
  - image\_threshold\_operations, 2170
- npptThreshold\_GTVal\_32f\_AC4R
  - image\_threshold\_operations, 2171
- npptThreshold\_GTVal\_32f\_C1IR
  - image\_threshold\_operations, 2171
- npptThreshold\_GTVal\_32f\_C1R
  - image\_threshold\_operations, 2172
- npptThreshold\_GTVal\_32f\_C3IR
  - image\_threshold\_operations, 2172
- npptThreshold\_GTVal\_32f\_C3R
  - image\_threshold\_operations, 2172
- npptThreshold\_GTVal\_8u\_AC4IR
  - image\_threshold\_operations, 2173
- npptThreshold\_GTVal\_8u\_AC4R
  - image\_threshold\_operations, 2173
- npptThreshold\_GTVal\_8u\_C1IR
  - image\_threshold\_operations, 2174
- npptThreshold\_GTVal\_8u\_C1R
  - image\_threshold\_operations, 2174
- npptThreshold\_GTVal\_8u\_C3IR
  - image\_threshold\_operations, 2175
- npptThreshold\_GTVal\_8u\_C3R
  - image\_threshold\_operations, 2175
- npptThreshold\_LT\_16s\_AC4IR
  - image\_threshold\_operations, 2175
- npptThreshold\_LT\_16s\_AC4R
  - image\_threshold\_operations, 2176
- npptThreshold\_LT\_16s\_C1IR
  - image\_threshold\_operations, 2176
- npptThreshold\_LT\_16s\_C1R
  - image\_threshold\_operations, 2177
- npptThreshold\_LT\_16s\_C3IR
  - image\_threshold\_operations, 2177
- npptThreshold\_LT\_16s\_C3R
  - image\_threshold\_operations, 2177
- npptThreshold\_LT\_16u\_AC4IR
  - image\_threshold\_operations, 2178
- npptThreshold\_LT\_16u\_AC4R
  - image\_threshold\_operations, 2178
- npptThreshold\_LT\_16u\_C1IR
  - image\_threshold\_operations, 2179
- npptThreshold\_LT\_16u\_C1R
  - image\_threshold\_operations, 2179
- npptThreshold\_LT\_16u\_C3IR
  - image\_threshold\_operations, 2179
- npptThreshold\_LT\_16u\_C3R
  - image\_threshold\_operations, 2179

image\_threshold\_operations, [2180](#)  
 nppiThreshold\_LT\_32f\_AC4IR  
     image\_threshold\_operations, [2180](#)  
 nppiThreshold\_LT\_32f\_AC4R  
     image\_threshold\_operations, [2181](#)  
 nppiThreshold\_LT\_32f\_C1IR  
     image\_threshold\_operations, [2181](#)  
 nppiThreshold\_LT\_32f\_C1R  
     image\_threshold\_operations, [2181](#)  
 nppiThreshold\_LT\_32f\_C3IR  
     image\_threshold\_operations, [2182](#)  
 nppiThreshold\_LT\_32f\_C3R  
     image\_threshold\_operations, [2182](#)  
 nppiThreshold\_LT\_8u\_AC4IR  
     image\_threshold\_operations, [2183](#)  
 nppiThreshold\_LT\_8u\_AC4R  
     image\_threshold\_operations, [2183](#)  
 nppiThreshold\_LT\_8u\_C1IR  
     image\_threshold\_operations, [2183](#)  
 nppiThreshold\_LT\_8u\_C1R  
     image\_threshold\_operations, [2184](#)  
 nppiThreshold\_LT\_8u\_C3IR  
     image\_threshold\_operations, [2184](#)  
 nppiThreshold\_LT\_8u\_C3R  
     image\_threshold\_operations, [2185](#)  
 nppiThreshold\_LTVaL\_16s\_AC4IR  
     image\_threshold\_operations, [2185](#)  
 nppiThreshold\_LTVaL\_16s\_AC4R  
     image\_threshold\_operations, [2185](#)  
 nppiThreshold\_LTVaL\_16s\_C1IR  
     image\_threshold\_operations, [2186](#)  
 nppiThreshold\_LTVaL\_16s\_C1R  
     image\_threshold\_operations, [2186](#)  
 nppiThreshold\_LTVaL\_16s\_C3IR  
     image\_threshold\_operations, [2187](#)  
 nppiThreshold\_LTVaL\_16s\_C3R  
     image\_threshold\_operations, [2187](#)  
 nppiThreshold\_LTVaL\_16u\_AC4IR  
     image\_threshold\_operations, [2188](#)  
 nppiThreshold\_LTVaL\_16u\_AC4R  
     image\_threshold\_operations, [2188](#)  
 nppiThreshold\_LTVaL\_16u\_C1IR  
     image\_threshold\_operations, [2188](#)  
 nppiThreshold\_LTVaL\_16u\_C1R  
     image\_threshold\_operations, [2189](#)  
 nppiThreshold\_LTVaL\_16u\_C3IR  
     image\_threshold\_operations, [2189](#)  
 nppiThreshold\_LTVaL\_16u\_C3R  
     image\_threshold\_operations, [2190](#)  
 nppiThreshold\_LTVaL\_32f\_AC4IR  
     image\_threshold\_operations, [2190](#)  
 nppiThreshold\_LTVaL\_32f\_AC4R  
     image\_threshold\_operations, [2190](#)  
 nppiThreshold\_LTVaL\_32f\_C1IR

image\_threshold\_operations, [2191](#)  
 nppiThreshold\_LTVaL\_32f\_C1R  
     image\_threshold\_operations, [2191](#)  
 nppiThreshold\_LTVaL\_32f\_C3IR  
     image\_threshold\_operations, [2192](#)  
 nppiThreshold\_LTVaL\_32f\_C3R  
     image\_threshold\_operations, [2192](#)  
 nppiThreshold\_LTVaL\_8u\_AC4IR  
     image\_threshold\_operations, [2193](#)  
 nppiThreshold\_LTVaL\_8u\_AC4R  
     image\_threshold\_operations, [2193](#)  
 nppiThreshold\_LTVaL\_8u\_C1IR  
     image\_threshold\_operations, [2193](#)  
 nppiThreshold\_LTVaL\_8u\_C1R  
     image\_threshold\_operations, [2194](#)  
 nppiThreshold\_LTVaL\_8u\_C3IR  
     image\_threshold\_operations, [2194](#)  
 nppiThreshold\_LTVaL\_8u\_C3R  
     image\_threshold\_operations, [2195](#)  
 nppiThreshold\_LTVaLGTVal\_16s\_AC4IR  
     image\_threshold\_operations, [2195](#)  
 nppiThreshold\_LTVaLGTVal\_16s\_AC4R  
     image\_threshold\_operations, [2196](#)  
 nppiThreshold\_LTVaLGTVal\_16s\_C1IR  
     image\_threshold\_operations, [2196](#)  
 nppiThreshold\_LTVaLGTVal\_16s\_C1R  
     image\_threshold\_operations, [2197](#)  
 nppiThreshold\_LTVaLGTVal\_16s\_C3IR  
     image\_threshold\_operations, [2197](#)  
 nppiThreshold\_LTVaLGTVal\_16s\_C3R  
     image\_threshold\_operations, [2198](#)  
 nppiThreshold\_LTVaLGTVal\_16u\_AC4IR  
     image\_threshold\_operations, [2198](#)  
 nppiThreshold\_LTVaLGTVal\_16u\_AC4R  
     image\_threshold\_operations, [2199](#)  
 nppiThreshold\_LTVaLGTVal\_16u\_C1IR  
     image\_threshold\_operations, [2199](#)  
 nppiThreshold\_LTVaLGTVal\_16u\_C1R  
     image\_threshold\_operations, [2200](#)  
 nppiThreshold\_LTVaLGTVal\_16u\_C3IR  
     image\_threshold\_operations, [2200](#)  
 nppiThreshold\_LTVaLGTVal\_16u\_C3R  
     image\_threshold\_operations, [2201](#)  
 nppiThreshold\_LTVaLGTVal\_32f\_AC4IR  
     image\_threshold\_operations, [2201](#)  
 nppiThreshold\_LTVaLGTVal\_32f\_AC4R  
     image\_threshold\_operations, [2202](#)  
 nppiThreshold\_LTVaLGTVal\_32f\_C1IR  
     image\_threshold\_operations, [2202](#)  
 nppiThreshold\_LTVaLGTVal\_32f\_C1R  
     image\_threshold\_operations, [2203](#)  
 nppiThreshold\_LTVaLGTVal\_32f\_C3IR  
     image\_threshold\_operations, [2203](#)  
 nppiThreshold\_LTVaLGTVal\_32f\_C3R



- image\_threshold\_operations, [2204](#)
- npptThreshold\_LTValGTVal\_8u\_AC4IR
  - image\_threshold\_operations, [2204](#)
- npptThreshold\_LTValGTVal\_8u\_AC4R
  - image\_threshold\_operations, [2205](#)
- npptThreshold\_LTValGTVal\_8u\_C1IR
  - image\_threshold\_operations, [2205](#)
- npptThreshold\_LTValGTVal\_8u\_C1R
  - image\_threshold\_operations, [2206](#)
- npptThreshold\_LTValGTVal\_8u\_C3IR
  - image\_threshold\_operations, [2206](#)
- npptThreshold\_LTValGTVal\_8u\_C3R
  - image\_threshold\_operations, [2207](#)
- npptThreshold\_Val\_16s\_AC4IR
  - image\_threshold\_operations, [2207](#)
- npptThreshold\_Val\_16s\_AC4R
  - image\_threshold\_operations, [2208](#)
- npptThreshold\_Val\_16s\_C1IR
  - image\_threshold\_operations, [2208](#)
- npptThreshold\_Val\_16s\_C1R
  - image\_threshold\_operations, [2209](#)
- npptThreshold\_Val\_16s\_C3IR
  - image\_threshold\_operations, [2209](#)
- npptThreshold\_Val\_16s\_C3R
  - image\_threshold\_operations, [2210](#)
- npptThreshold\_Val\_16u\_AC4IR
  - image\_threshold\_operations, [2210](#)
- npptThreshold\_Val\_16u\_AC4R
  - image\_threshold\_operations, [2211](#)
- npptThreshold\_Val\_16u\_C1IR
  - image\_threshold\_operations, [2211](#)
- npptThreshold\_Val\_16u\_C1R
  - image\_threshold\_operations, [2212](#)
- npptThreshold\_Val\_16u\_C3IR
  - image\_threshold\_operations, [2212](#)
- npptThreshold\_Val\_16u\_C3R
  - image\_threshold\_operations, [2213](#)
- npptThreshold\_Val\_32f\_AC4IR
  - image\_threshold\_operations, [2213](#)
- npptThreshold\_Val\_32f\_AC4R
  - image\_threshold\_operations, [2214](#)
- npptThreshold\_Val\_32f\_C1IR
  - image\_threshold\_operations, [2214](#)
- npptThreshold\_Val\_32f\_C1R
  - image\_threshold\_operations, [2215](#)
- npptThreshold\_Val\_32f\_C3IR
  - image\_threshold\_operations, [2215](#)
- npptThreshold\_Val\_32f\_C3R
  - image\_threshold\_operations, [2216](#)
- npptThreshold\_Val\_8u\_AC4IR
  - image\_threshold\_operations, [2216](#)
- npptThreshold\_Val\_8u\_AC4R
  - image\_threshold\_operations, [2217](#)
- npptThreshold\_Val\_8u\_C1IR
  - image\_threshold\_operations, [2217](#)
- npptThreshold\_Val\_8u\_C1R
  - image\_threshold\_operations, [2218](#)
- npptThreshold\_Val\_8u\_C3IR
  - image\_threshold\_operations, [2218](#)
- npptThreshold\_Val\_8u\_C3R
  - image\_threshold\_operations, [2219](#)
- npptTranspose\_16s\_C1R
  - image\_transpose, [917](#)
- npptTranspose\_16s\_C3R
  - image\_transpose, [917](#)
- npptTranspose\_16s\_C4R
  - image\_transpose, [918](#)
- npptTranspose\_16u\_C1R
  - image\_transpose, [918](#)
- npptTranspose\_16u\_C3R
  - image\_transpose, [918](#)
- npptTranspose\_16u\_C4R
  - image\_transpose, [919](#)
- npptTranspose\_32f\_C1R
  - image\_transpose, [919](#)
- npptTranspose\_32f\_C3R
  - image\_transpose, [919](#)
- npptTranspose\_32f\_C4R
  - image\_transpose, [920](#)
- npptTranspose\_32s\_C1R
  - image\_transpose, [920](#)
- npptTranspose\_32s\_C3R
  - image\_transpose, [920](#)
- npptTranspose\_32s\_C4R
  - image\_transpose, [921](#)
- npptTranspose\_8u\_C1R
  - image\_transpose, [921](#)
- npptTranspose\_8u\_C3R
  - image\_transpose, [921](#)
- npptTranspose\_8u\_C4R
  - image\_transpose, [922](#)
- npptValidNormLevelGetBufferHostSize\_16u32f\_-AC4R
  - crosscorrvalidnormlevel, [2008](#)
- npptValidNormLevelGetBufferHostSize\_16u32f\_-C1R
  - crosscorrvalidnormlevel, [2009](#)
- npptValidNormLevelGetBufferHostSize\_16u32f\_-C3R
  - crosscorrvalidnormlevel, [2009](#)
- npptValidNormLevelGetBufferHostSize\_16u32f\_-C4R
  - crosscorrvalidnormlevel, [2009](#)
- npptValidNormLevelGetBufferHostSize\_32f\_-AC4R
  - crosscorrvalidnormlevel, [2010](#)
- npptValidNormLevelGetBufferHostSize\_32f\_C1R
  - crosscorrvalidnormlevel, [2010](#)

- crosscorrvalidnormlevel, [2010](#)
- nppiValidNormLevelGetBufferHostSize\_32f\_C4R
  - crosscorrvalidnormlevel, [2010](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_-AC4R
  - crosscorrvalidnormlevel, [2011](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_-C1R
  - crosscorrvalidnormlevel, [2011](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_-C3R
  - crosscorrvalidnormlevel, [2011](#)
- nppiValidNormLevelGetBufferHostSize\_8s32f\_-C4R
  - crosscorrvalidnormlevel, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_-AC4R
  - crosscorrvalidnormlevel, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_-C1R
  - crosscorrvalidnormlevel, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_-C3R
  - crosscorrvalidnormlevel, [2012](#)
- nppiValidNormLevelGetBufferHostSize\_8u32f\_-C4R
  - crosscorrvalidnormlevel, [2013](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-AC4RSfs
  - crosscorrvalidnormlevel, [2013](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-C1RSfs
  - crosscorrvalidnormlevel, [2013](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-C3RSfs
  - crosscorrvalidnormlevel, [2014](#)
- nppiValidNormLevelGetBufferHostSize\_8u\_-C4RSfs
  - crosscorrvalidnormlevel, [2014](#)
- nppiWarpAffine\_16u\_AC4R
  - image\_affine\_transform, [1277](#)
- nppiWarpAffine\_16u\_C1R
  - image\_affine\_transform, [1278](#)
- nppiWarpAffine\_16u\_C3R
  - image\_affine\_transform, [1278](#)
- nppiWarpAffine\_16u\_C4R
  - image\_affine\_transform, [1279](#)
- nppiWarpAffine\_16u\_P3R
  - image\_affine\_transform, [1279](#)
- nppiWarpAffine\_16u\_P4R
  - image\_affine\_transform, [1280](#)
- nppiWarpAffine\_32f\_AC4R
  - image\_affine\_transform, [1280](#)
- nppiWarpAffine\_32f\_C1R
  - image\_affine\_transform, [1281](#)
- nppiWarpAffine\_32f\_C3R
  - image\_affine\_transform, [1281](#)
- nppiWarpAffine\_32f\_C4R
  - image\_affine\_transform, [1282](#)
- nppiWarpAffine\_32f\_P3R
  - image\_affine\_transform, [1282](#)
- nppiWarpAffine\_32f\_P4R
  - image\_affine\_transform, [1283](#)
- nppiWarpAffine\_32s\_AC4R
  - image\_affine\_transform, [1283](#)
- nppiWarpAffine\_32s\_C1R
  - image\_affine\_transform, [1284](#)
- nppiWarpAffine\_32s\_C3R
  - image\_affine\_transform, [1284](#)
- nppiWarpAffine\_32s\_C4R
  - image\_affine\_transform, [1285](#)
- nppiWarpAffine\_32s\_P3R
  - image\_affine\_transform, [1285](#)
- nppiWarpAffine\_32s\_P4R
  - image\_affine\_transform, [1286](#)
- nppiWarpAffine\_64f\_AC4R
  - image\_affine\_transform, [1286](#)
- nppiWarpAffine\_64f\_C1R
  - image\_affine\_transform, [1287](#)
- nppiWarpAffine\_64f\_C3R
  - image\_affine\_transform, [1287](#)
- nppiWarpAffine\_64f\_C4R
  - image\_affine\_transform, [1288](#)
- nppiWarpAffine\_64f\_P3R
  - image\_affine\_transform, [1288](#)
- nppiWarpAffine\_64f\_P4R
  - image\_affine\_transform, [1289](#)
- nppiWarpAffine\_8u\_AC4R
  - image\_affine\_transform, [1289](#)
- nppiWarpAffine\_8u\_C1R
  - image\_affine\_transform, [1290](#)
- nppiWarpAffine\_8u\_C3R
  - image\_affine\_transform, [1290](#)
- nppiWarpAffine\_8u\_C4R
  - image\_affine\_transform, [1291](#)
- nppiWarpAffine\_8u\_P3R
  - image\_affine\_transform, [1291](#)
- nppiWarpAffine\_8u\_P4R
  - image\_affine\_transform, [1292](#)
- nppiWarpAffineBack\_16u\_AC4R
  - image\_affine\_transform, [1292](#)
- nppiWarpAffineBack\_16u\_C1R
  - image\_affine\_transform, [1293](#)
- nppiWarpAffineBack\_16u\_C3R
  - image\_affine\_transform, [1293](#)
- nppiWarpAffineBack\_16u\_C4R
  - image\_affine\_transform, [1294](#)



- nppiWarpAffineBack\_16u\_P3R
  - image\_affine\_transform, [1294](#)
- nppiWarpAffineBack\_16u\_P4R
  - image\_affine\_transform, [1295](#)
- nppiWarpAffineBack\_32f\_AC4R
  - image\_affine\_transform, [1295](#)
- nppiWarpAffineBack\_32f\_C1R
  - image\_affine\_transform, [1296](#)
- nppiWarpAffineBack\_32f\_C3R
  - image\_affine\_transform, [1296](#)
- nppiWarpAffineBack\_32f\_C4R
  - image\_affine\_transform, [1297](#)
- nppiWarpAffineBack\_32f\_P3R
  - image\_affine\_transform, [1297](#)
- nppiWarpAffineBack\_32f\_P4R
  - image\_affine\_transform, [1298](#)
- nppiWarpAffineBack\_32s\_AC4R
  - image\_affine\_transform, [1298](#)
- nppiWarpAffineBack\_32s\_C1R
  - image\_affine\_transform, [1299](#)
- nppiWarpAffineBack\_32s\_C3R
  - image\_affine\_transform, [1299](#)
- nppiWarpAffineBack\_32s\_C4R
  - image\_affine\_transform, [1300](#)
- nppiWarpAffineBack\_32s\_P3R
  - image\_affine\_transform, [1300](#)
- nppiWarpAffineBack\_32s\_P4R
  - image\_affine\_transform, [1301](#)
- nppiWarpAffineBack\_8u\_AC4R
  - image\_affine\_transform, [1301](#)
- nppiWarpAffineBack\_8u\_C1R
  - image\_affine\_transform, [1302](#)
- nppiWarpAffineBack\_8u\_C3R
  - image\_affine\_transform, [1302](#)
- nppiWarpAffineBack\_8u\_C4R
  - image\_affine\_transform, [1303](#)
- nppiWarpAffineBack\_8u\_P3R
  - image\_affine\_transform, [1303](#)
- nppiWarpAffineBack\_8u\_P4R
  - image\_affine\_transform, [1304](#)
- nppiWarpAffineQuad\_16u\_AC4R
  - image\_affine\_transform, [1304](#)
- nppiWarpAffineQuad\_16u\_C1R
  - image\_affine\_transform, [1305](#)
- nppiWarpAffineQuad\_16u\_C3R
  - image\_affine\_transform, [1305](#)
- nppiWarpAffineQuad\_16u\_C4R
  - image\_affine\_transform, [1306](#)
- nppiWarpAffineQuad\_16u\_P3R
  - image\_affine\_transform, [1306](#)
- nppiWarpAffineQuad\_16u\_P4R
  - image\_affine\_transform, [1307](#)
- nppiWarpAffineQuad\_32f\_AC4R
  - image\_affine\_transform, [1307](#)
- nppiWarpAffineQuad\_32f\_C1R
  - image\_affine\_transform, [1308](#)
- nppiWarpAffineQuad\_32f\_C3R
  - image\_affine\_transform, [1308](#)
- nppiWarpAffineQuad\_32f\_C4R
  - image\_affine\_transform, [1309](#)
- nppiWarpAffineQuad\_32f\_P3R
  - image\_affine\_transform, [1309](#)
- nppiWarpAffineQuad\_32f\_P4R
  - image\_affine\_transform, [1310](#)
- nppiWarpAffineQuad\_32s\_AC4R
  - image\_affine\_transform, [1310](#)
- nppiWarpAffineQuad\_32s\_C1R
  - image\_affine\_transform, [1311](#)
- nppiWarpAffineQuad\_32s\_C3R
  - image\_affine\_transform, [1311](#)
- nppiWarpAffineQuad\_32s\_C4R
  - image\_affine\_transform, [1312](#)
- nppiWarpAffineQuad\_32s\_P3R
  - image\_affine\_transform, [1312](#)
- nppiWarpAffineQuad\_32s\_P4R
  - image\_affine\_transform, [1313](#)
- nppiWarpAffineQuad\_8u\_AC4R
  - image\_affine\_transform, [1313](#)
- nppiWarpAffineQuad\_8u\_C1R
  - image\_affine\_transform, [1314](#)
- nppiWarpAffineQuad\_8u\_C3R
  - image\_affine\_transform, [1314](#)
- nppiWarpAffineQuad\_8u\_C4R
  - image\_affine\_transform, [1315](#)
- nppiWarpAffineQuad\_8u\_P3R
  - image\_affine\_transform, [1315](#)
- nppiWarpAffineQuad\_8u\_P4R
  - image\_affine\_transform, [1316](#)
- nppiWarpPerspective\_16u\_AC4R
  - image\_perspective\_transforms, [1326](#)
- nppiWarpPerspective\_16u\_C1R
  - image\_perspective\_transforms, [1327](#)
- nppiWarpPerspective\_16u\_C3R
  - image\_perspective\_transforms, [1327](#)
- nppiWarpPerspective\_16u\_C4R
  - image\_perspective\_transforms, [1328](#)
- nppiWarpPerspective\_16u\_P3R
  - image\_perspective\_transforms, [1328](#)
- nppiWarpPerspective\_16u\_P4R
  - image\_perspective\_transforms, [1329](#)
- nppiWarpPerspective\_32f\_AC4R
  - image\_perspective\_transforms, [1329](#)
- nppiWarpPerspective\_32f\_C1R
  - image\_perspective\_transforms, [1330](#)
- nppiWarpPerspective\_32f\_C3R
  - image\_perspective\_transforms, [1330](#)
- nppiWarpPerspective\_32f\_C4R
  - image\_perspective\_transforms, [1331](#)

- npPiWarpPerspective\_32f\_P3R
  - image\_perspective\_transforms, [1331](#)
- npPiWarpPerspective\_32f\_P4R
  - image\_perspective\_transforms, [1332](#)
- npPiWarpPerspective\_32s\_AC4R
  - image\_perspective\_transforms, [1332](#)
- npPiWarpPerspective\_32s\_C1R
  - image\_perspective\_transforms, [1333](#)
- npPiWarpPerspective\_32s\_C3R
  - image\_perspective\_transforms, [1333](#)
- npPiWarpPerspective\_32s\_C4R
  - image\_perspective\_transforms, [1334](#)
- npPiWarpPerspective\_32s\_P3R
  - image\_perspective\_transforms, [1334](#)
- npPiWarpPerspective\_32s\_P4R
  - image\_perspective\_transforms, [1334](#)
- npPiWarpPerspective\_8u\_AC4R
  - image\_perspective\_transforms, [1335](#)
- npPiWarpPerspective\_8u\_C1R
  - image\_perspective\_transforms, [1335](#)
- npPiWarpPerspective\_8u\_C3R
  - image\_perspective\_transforms, [1336](#)
- npPiWarpPerspective\_8u\_C4R
  - image\_perspective\_transforms, [1336](#)
- npPiWarpPerspective\_8u\_P3R
  - image\_perspective\_transforms, [1337](#)
- npPiWarpPerspective\_8u\_P4R
  - image\_perspective\_transforms, [1337](#)
- npPiWarpPerspectiveBack\_16u\_AC4R
  - image\_perspective\_transforms, [1338](#)
- npPiWarpPerspectiveBack\_16u\_C1R
  - image\_perspective\_transforms, [1338](#)
- npPiWarpPerspectiveBack\_16u\_C3R
  - image\_perspective\_transforms, [1339](#)
- npPiWarpPerspectiveBack\_16u\_C4R
  - image\_perspective\_transforms, [1339](#)
- npPiWarpPerspectiveBack\_16u\_P3R
  - image\_perspective\_transforms, [1340](#)
- npPiWarpPerspectiveBack\_16u\_P4R
  - image\_perspective\_transforms, [1340](#)
- npPiWarpPerspectiveBack\_32f\_AC4R
  - image\_perspective\_transforms, [1341](#)
- npPiWarpPerspectiveBack\_32f\_C1R
  - image\_perspective\_transforms, [1341](#)
- npPiWarpPerspectiveBack\_32f\_C3R
  - image\_perspective\_transforms, [1342](#)
- npPiWarpPerspectiveBack\_32f\_C4R
  - image\_perspective\_transforms, [1342](#)
- npPiWarpPerspectiveBack\_32f\_P3R
  - image\_perspective\_transforms, [1343](#)
- npPiWarpPerspectiveBack\_32f\_P4R
  - image\_perspective\_transforms, [1343](#)
- npPiWarpPerspectiveBack\_32s\_AC4R
  - image\_perspective\_transforms, [1344](#)
- npPiWarpPerspectiveBack\_32s\_C1R
  - image\_perspective\_transforms, [1344](#)
- npPiWarpPerspectiveBack\_32s\_C3R
  - image\_perspective\_transforms, [1345](#)
- npPiWarpPerspectiveBack\_32s\_C4R
  - image\_perspective\_transforms, [1345](#)
- npPiWarpPerspectiveBack\_32s\_P3R
  - image\_perspective\_transforms, [1346](#)
- npPiWarpPerspectiveBack\_32s\_P4R
  - image\_perspective\_transforms, [1346](#)
- npPiWarpPerspectiveBack\_8u\_AC4R
  - image\_perspective\_transforms, [1347](#)
- npPiWarpPerspectiveBack\_8u\_C1R
  - image\_perspective\_transforms, [1347](#)
- npPiWarpPerspectiveBack\_8u\_C3R
  - image\_perspective\_transforms, [1348](#)
- npPiWarpPerspectiveBack\_8u\_C4R
  - image\_perspective\_transforms, [1348](#)
- npPiWarpPerspectiveBack\_8u\_P3R
  - image\_perspective\_transforms, [1349](#)
- npPiWarpPerspectiveBack\_8u\_P4R
  - image\_perspective\_transforms, [1349](#)
- npPiWarpPerspectiveQuad\_16u\_AC4R
  - image\_perspective\_transforms, [1350](#)
- npPiWarpPerspectiveQuad\_16u\_C1R
  - image\_perspective\_transforms, [1350](#)
- npPiWarpPerspectiveQuad\_16u\_C3R
  - image\_perspective\_transforms, [1351](#)
- npPiWarpPerspectiveQuad\_16u\_C4R
  - image\_perspective\_transforms, [1351](#)
- npPiWarpPerspectiveQuad\_16u\_P3R
  - image\_perspective\_transforms, [1352](#)
- npPiWarpPerspectiveQuad\_16u\_P4R
  - image\_perspective\_transforms, [1352](#)
- npPiWarpPerspectiveQuad\_32f\_AC4R
  - image\_perspective\_transforms, [1353](#)
- npPiWarpPerspectiveQuad\_32f\_C1R
  - image\_perspective\_transforms, [1353](#)
- npPiWarpPerspectiveQuad\_32f\_C3R
  - image\_perspective\_transforms, [1354](#)
- npPiWarpPerspectiveQuad\_32f\_C4R
  - image\_perspective\_transforms, [1354](#)
- npPiWarpPerspectiveQuad\_32f\_P3R
  - image\_perspective\_transforms, [1355](#)
- npPiWarpPerspectiveQuad\_32f\_P4R
  - image\_perspective\_transforms, [1355](#)
- npPiWarpPerspectiveQuad\_32s\_AC4R
  - image\_perspective\_transforms, [1356](#)
- npPiWarpPerspectiveQuad\_32s\_C1R
  - image\_perspective\_transforms, [1356](#)
- npPiWarpPerspectiveQuad\_32s\_C3R
  - image\_perspective\_transforms, [1357](#)
- npPiWarpPerspectiveQuad\_32s\_C4R
  - image\_perspective\_transforms, [1357](#)

- npapiWarpPerspectiveQuad\_32s\_P3R
  - image\_perspective\_transforms, [1358](#)
- npapiWarpPerspectiveQuad\_32s\_P4R
  - image\_perspective\_transforms, [1358](#)
- npapiWarpPerspectiveQuad\_8u\_AC4R
  - image\_perspective\_transforms, [1359](#)
- npapiWarpPerspectiveQuad\_8u\_C1R
  - image\_perspective\_transforms, [1359](#)
- npapiWarpPerspectiveQuad\_8u\_C3R
  - image\_perspective\_transforms, [1360](#)
- npapiWarpPerspectiveQuad\_8u\_C4R
  - image\_perspective\_transforms, [1360](#)
- npapiWarpPerspectiveQuad\_8u\_P3R
  - image\_perspective\_transforms, [1361](#)
- npapiWarpPerspectiveQuad\_8u\_P4R
  - image\_perspective\_transforms, [1361](#)
- npapiXor\_16u\_AC4IR
  - image\_xor, [458](#)
- npapiXor\_16u\_AC4R
  - image\_xor, [458](#)
- npapiXor\_16u\_C1IR
  - image\_xor, [458](#)
- npapiXor\_16u\_C1R
  - image\_xor, [459](#)
- npapiXor\_16u\_C3IR
  - image\_xor, [459](#)
- npapiXor\_16u\_C3R
  - image\_xor, [459](#)
- npapiXor\_16u\_C4IR
  - image\_xor, [460](#)
- npapiXor\_16u\_C4R
  - image\_xor, [460](#)
- npapiXor\_32s\_AC4IR
  - image\_xor, [461](#)
- npapiXor\_32s\_AC4R
  - image\_xor, [461](#)
- npapiXor\_32s\_C1IR
  - image\_xor, [461](#)
- npapiXor\_32s\_C1R
  - image\_xor, [462](#)
- npapiXor\_32s\_C3IR
  - image\_xor, [462](#)
- npapiXor\_32s\_C3R
  - image\_xor, [462](#)
- npapiXor\_32s\_C4IR
  - image\_xor, [463](#)
- npapiXor\_32s\_C4R
  - image\_xor, [463](#)
- npapiXor\_8u\_AC4IR
  - image\_xor, [464](#)
- npapiXor\_8u\_AC4R
  - image\_xor, [464](#)
- npapiXor\_8u\_C1IR
  - image\_xor, [464](#)
- npapiXor\_8u\_C1R
  - image\_xor, [465](#)
- npapiXor\_8u\_C3IR
  - image\_xor, [465](#)
- npapiXor\_8u\_C3R
  - image\_xor, [465](#)
- npapiXor\_8u\_C4IR
  - image\_xor, [466](#)
- npapiXor\_8u\_C4R
  - image\_xor, [466](#)
- npapiXorC\_16u\_AC4IR
  - image\_xorc, [395](#)
- npapiXorC\_16u\_AC4R
  - image\_xorc, [395](#)
- npapiXorC\_16u\_C1IR
  - image\_xorc, [395](#)
- npapiXorC\_16u\_C1R
  - image\_xorc, [396](#)
- npapiXorC\_16u\_C3IR
  - image\_xorc, [396](#)
- npapiXorC\_16u\_C3R
  - image\_xorc, [396](#)
- npapiXorC\_16u\_C4IR
  - image\_xorc, [397](#)
- npapiXorC\_16u\_C4R
  - image\_xorc, [397](#)
- npapiXorC\_32s\_AC4IR
  - image\_xorc, [397](#)
- npapiXorC\_32s\_AC4R
  - image\_xorc, [398](#)
- npapiXorC\_32s\_C1IR
  - image\_xorc, [398](#)
- npapiXorC\_32s\_C1R
  - image\_xorc, [398](#)
- npapiXorC\_32s\_C3IR
  - image\_xorc, [399](#)
- npapiXorC\_32s\_C3R
  - image\_xorc, [399](#)
- npapiXorC\_32s\_C4IR
  - image\_xorc, [399](#)
- npapiXorC\_32s\_C4R
  - image\_xorc, [400](#)
- npapiXorC\_8u\_AC4IR
  - image\_xorc, [400](#)
- npapiXorC\_8u\_AC4R
  - image\_xorc, [400](#)
- npapiXorC\_8u\_C1IR
  - image\_xorc, [401](#)
- npapiXorC\_8u\_C1R
  - image\_xorc, [401](#)
- npapiXorC\_8u\_C3IR
  - image\_xorc, [401](#)
- npapiXorC\_8u\_C3R
  - image\_xorc, [402](#)

- nppiXorC\_8u\_C4IR
  - image\_xorc, [402](#)
- nppiXorC\_8u\_C4R
  - image\_xorc, [402](#)
- nppiXYZToRGB\_8u\_AC4R
  - image\_color\_model\_conversion, [562](#)
- nppiXYZToRGB\_8u\_C3R
  - image\_color\_model\_conversion, [562](#)
- nppiYCbCr411\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [588](#)
- nppiYCbCr411\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [588](#)
- nppiYCbCr411ToBGR\_8u\_P3C3R
  - image\_color\_model\_conversion, [562](#)
- nppiYCbCr411ToBGR\_8u\_P3C4R
  - image\_color\_model\_conversion, [563](#)
- nppiYCbCr411ToYCbCr420\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [588](#)
- nppiYCbCr411ToYCbCr420\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [589](#)
- nppiYCbCr411ToYCbCr420\_8u\_P3R
  - image\_color\_sampling\_format\_conversion, [589](#)
- nppiYCbCr411ToYCbCr422\_8u\_P2C2R
  - image\_color\_sampling\_format\_conversion, [590](#)
- nppiYCbCr411ToYCbCr422\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [590](#)
- nppiYCbCr411ToYCbCr422\_8u\_P3C2R
  - image\_color\_sampling\_format\_conversion, [590](#)
- nppiYCbCr411ToYCbCr422\_8u\_P3R
  - image\_color\_sampling\_format\_conversion, [591](#)
- nppiYCbCr411ToYCrCb420\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [591](#)
- nppiYCbCr411ToYCrCb422\_8u\_P3C2R
  - image\_color\_sampling\_format\_conversion, [592](#)
- nppiYCbCr411ToYCrCb422\_8u\_P3R
  - image\_color\_sampling\_format\_conversion, [592](#)
- nppiYCbCr420\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [592](#)
- nppiYCbCr420\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [593](#)
- nppiYCbCr420ToBGR\_709CSC\_8u\_P3C3R
  - image\_color\_model\_conversion, [563](#)
- nppiYCbCr420ToBGR\_709HDTV\_8u\_P3C4R
  - image\_color\_model\_conversion, [563](#)
- nppiYCbCr420ToBGR\_8u\_P3C3R
  - image\_color\_model\_conversion, [564](#)
- nppiYCbCr420ToBGR\_8u\_P3C4R
  - image\_color\_model\_conversion, [564](#)
- nppiYCbCr420ToCbYCr422\_8u\_P2C2R
  - image\_color\_sampling\_format\_conversion, [593](#)
- nppiYCbCr420ToRGB\_8u\_P3C3R
  - image\_color\_model\_conversion, [565](#)
- nppiYCbCr420ToYCbCr411\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [594](#)
- nppiYCbCr420ToYCbCr411\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [594](#)
- nppiYCbCr420ToYCbCr422\_8u\_P2C2R
  - image\_color\_sampling\_format\_conversion, [595](#)
- nppiYCbCr420ToYCbCr422\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [595](#)
- nppiYCbCr420ToYCbCr422\_8u\_P3R
  - image\_color\_sampling\_format\_conversion, [595](#)
- nppiYCbCr420ToYCrCb420\_8u\_P2P3R
  - image\_color\_sampling\_format\_conversion, [596](#)
- nppiYCbCr422\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, [596](#)
- nppiYCbCr422\_8u\_P3C2R
  - image\_color\_sampling\_format\_conversion, [597](#)
- nppiYCbCr422ToBGR\_8u\_C2C3R
  - image\_color\_model\_conversion, [565](#)
- nppiYCbCr422ToBGR\_8u\_C2C4R
  - image\_color\_model\_conversion, [565](#)
- nppiYCbCr422ToBGR\_8u\_P3C3R
  - image\_color\_model\_conversion, [566](#)
- nppiYCbCr422ToCbYCr422\_8u\_C2R
  - image\_color\_sampling\_format\_conversion, [597](#)
- nppiYCbCr422ToRGB\_8u\_C2C3R
  - image\_color\_model\_conversion, [566](#)
- nppiYCbCr422ToRGB\_8u\_C2P3R
  - image\_color\_model\_conversion, [566](#)
- nppiYCbCr422ToRGB\_8u\_P3C3R
  - image\_color\_model\_conversion, [567](#)
- nppiYCbCr422ToYCbCr411\_8u\_C2P2R

- image\_color\_sampling\_format\_conversion, [597](#)
- npYCbCr422ToYCbCr411\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, [598](#)
- npYCbCr422ToYCbCr411\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [598](#)
- npYCbCr422ToYCbCr411\_8u\_P3R
  - image\_color\_sampling\_format\_conversion, [599](#)
- npYCbCr422ToYCbCr420\_8u\_C2P2R
  - image\_color\_sampling\_format\_conversion, [599](#)
- npYCbCr422ToYCbCr420\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, [600](#)
- npYCbCr422ToYCbCr420\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [600](#)
- npYCbCr422ToYCbCr420\_8u\_P3R
  - image\_color\_sampling\_format\_conversion, [600](#)
- npYCbCr422ToYCrCb420\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, [601](#)
- npYCbCr422ToYCrCb422\_8u\_C2R
  - image\_color\_sampling\_format\_conversion, [601](#)
- npYCbCr422ToYCrCb422\_8u\_P3C2R
  - image\_color\_sampling\_format\_conversion, [602](#)
- npYCbCrToBGR\_709CSC\_8u\_P3C3R
  - image\_color\_model\_conversion, [567](#)
- npYCbCrToBGR\_709CSC\_8u\_P3C4R
  - image\_color\_model\_conversion, [567](#)
- npYCbCrToBGR\_8u\_P3C3R
  - image\_color\_model\_conversion, [568](#)
- npYCbCrToBGR\_8u\_P3C4R
  - image\_color\_model\_conversion, [568](#)
- npYCbCrToRGB\_8u\_AC4R
  - image\_color\_model\_conversion, [569](#)
- npYCbCrToRGB\_8u\_C3R
  - image\_color\_model\_conversion, [569](#)
- npYCbCrToRGB\_8u\_P3C3R
  - image\_color\_model\_conversion, [569](#)
- npYCbCrToRGB\_8u\_P3C4R
  - image\_color\_model\_conversion, [570](#)
- npYCbCrToRGB\_8u\_P3R
  - image\_color\_model\_conversion, [570](#)
- npYCCToRGB\_8u\_AC4R
  - image\_color\_model\_conversion, [570](#)
- npYCCToRGB\_8u\_C3R
  - image\_color\_model\_conversion, [571](#)
- npYCrCb420ToCbYCr422\_8u\_P3C2R
  - image\_color\_sampling\_format\_conversion, [602](#)
- npYCrCb420ToRGB\_8u\_P3C4R
  - image\_color\_model\_conversion, [571](#)
- npYCrCb420ToYCbCr411\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [602](#)
- npYCrCb420ToYCbCr420\_8u\_P3P2R
  - image\_color\_sampling\_format\_conversion, [603](#)
- npYCrCb420ToYCbCr422\_8u\_P3C2R
  - image\_color\_sampling\_format\_conversion, [603](#)
- npYCrCb420ToYCbCr422\_8u\_P3R
  - image\_color\_sampling\_format\_conversion, [604](#)
- npYCrCb422ToRGB\_8u\_C2C3R
  - image\_color\_model\_conversion, [571](#)
- npYCrCb422ToRGB\_8u\_C2P3R
  - image\_color\_model\_conversion, [572](#)
- npYCrCb422ToYCbCr411\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, [604](#)
- npYCrCb422ToYCbCr420\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, [605](#)
- npYCrCb422ToYCbCr422\_8u\_C2P3R
  - image\_color\_sampling\_format\_conversion, [605](#)
- npYUV420ToBGR\_8u\_P3C3R
  - image\_color\_model\_conversion, [572](#)
- npYUV420ToRGB\_8u\_P3AC4R
  - image\_color\_model\_conversion, [572](#)
- npYUV420ToRGB\_8u\_P3C3R
  - image\_color\_model\_conversion, [573](#)
- npYUV420ToRGB\_8u\_P3R
  - image\_color\_model\_conversion, [573](#)
- npYUV422ToRGB\_8u\_C2C3R
  - image\_color\_model\_conversion, [573](#)
- npYUV422ToRGB\_8u\_P3AC4R
  - image\_color\_model\_conversion, [574](#)
- npYUV422ToRGB\_8u\_P3C3R
  - image\_color\_model\_conversion, [574](#)
- npYUV422ToRGB\_8u\_P3R
  - image\_color\_model\_conversion, [574](#)
- npYUVToBGR\_8u\_AC4R
  - image\_color\_model\_conversion, [575](#)
- npYUVToBGR\_8u\_C3R
  - image\_color\_model\_conversion, [575](#)
- npYUVToBGR\_8u\_P3C3R
  - image\_color\_model\_conversion, [575](#)
- npYUVToBGR\_8u\_P3R
  - image\_color\_model\_conversion, [576](#)

- nppiYUVToRGB\_8u\_AC4R
  - image\_color\_model\_conversion, [576](#)
- nppiYUVToRGB\_8u\_C3R
  - image\_color\_model\_conversion, [576](#)
- nppiYUVToRGB\_8u\_P3C3R
  - image\_color\_model\_conversion, [577](#)
- nppiYUVToRGB\_8u\_P3R
  - image\_color\_model\_conversion, [577](#)
- NppLibraryVersion, [2634](#)
  - build, [2634](#)
  - major, [2634](#)
  - minor, [2634](#)
- NppRoundMode
  - typedefs\_npp, [43](#)
- npps10Log10\_32s\_ISfs
  - signal\_10log10, [2370](#)
- npps10Log10\_32s\_Sfs
  - signal\_10log10, [2370](#)
- nppsAbs\_16s
  - signal\_abs, [2344](#)
- nppsAbs\_16s\_I
  - signal\_abs, [2344](#)
- nppsAbs\_32f
  - signal\_abs, [2345](#)
- nppsAbs\_32f\_I
  - signal\_abs, [2345](#)
- nppsAbs\_32s
  - signal\_abs, [2345](#)
- nppsAbs\_32s\_I
  - signal\_abs, [2345](#)
- nppsAbs\_64f
  - signal\_abs, [2346](#)
- nppsAbs\_64f\_I
  - signal\_abs, [2346](#)
- nppsAdd\_16s
  - signal\_add, [2296](#)
- nppsAdd\_16s32f
  - signal\_add, [2296](#)
- nppsAdd\_16s32s\_I
  - signal\_add, [2296](#)
- nppsAdd\_16s\_I
  - signal\_add, [2297](#)
- nppsAdd\_16s\_ISfs
  - signal\_add, [2297](#)
- nppsAdd\_16s\_Sfs
  - signal\_add, [2297](#)
- nppsAdd\_16sc\_ISfs
  - signal\_add, [2298](#)
- nppsAdd\_16sc\_Sfs
  - signal\_add, [2298](#)
- nppsAdd\_16u
  - signal\_add, [2298](#)
- nppsAdd\_16u\_ISfs
  - signal\_add, [2299](#)
- nppsAdd\_16u\_Sfs
  - signal\_add, [2299](#)
- nppsAdd\_32f
  - signal\_add, [2299](#)
- nppsAdd\_32f\_I
  - signal\_add, [2300](#)
- nppsAdd\_32fc
  - signal\_add, [2300](#)
- nppsAdd\_32fc\_I
  - signal\_add, [2300](#)
- nppsAdd\_32s\_ISfs
  - signal\_add, [2301](#)
- nppsAdd\_32s\_Sfs
  - signal\_add, [2301](#)
- nppsAdd\_32sc\_ISfs
  - signal\_add, [2301](#)
- nppsAdd\_32sc\_Sfs
  - signal\_add, [2302](#)
- nppsAdd\_32u
  - signal\_add, [2302](#)
- nppsAdd\_64f
  - signal\_add, [2302](#)
- nppsAdd\_64f\_I
  - signal\_add, [2303](#)
- nppsAdd\_64fc
  - signal\_add, [2303](#)
- nppsAdd\_64fc\_I
  - signal\_add, [2303](#)
- nppsAdd\_64s\_Sfs
  - signal\_add, [2304](#)
- nppsAdd\_8u16u
  - signal\_add, [2304](#)
- nppsAdd\_8u\_ISfs
  - signal\_add, [2304](#)
- nppsAdd\_8u\_Sfs
  - signal\_add, [2305](#)
- nppsAddC\_16s\_ISfs
  - signal\_addc, [2248](#)
- nppsAddC\_16s\_Sfs
  - signal\_addc, [2248](#)
- nppsAddC\_16sc\_ISfs
  - signal\_addc, [2249](#)
- nppsAddC\_16sc\_Sfs
  - signal\_addc, [2249](#)
- nppsAddC\_16u\_ISfs
  - signal\_addc, [2249](#)
- nppsAddC\_16u\_Sfs
  - signal\_addc, [2250](#)
- nppsAddC\_32f
  - signal\_addc, [2250](#)
- nppsAddC\_32f\_I
  - signal\_addc, [2250](#)
- nppsAddC\_32fc
  - signal\_addc, [2251](#)



- nppsAddC\_32fc\_I
  - signal\_addc, [2251](#)
- nppsAddC\_32s\_ISfs
  - signal\_addc, [2251](#)
- nppsAddC\_32s\_Sfs
  - signal\_addc, [2252](#)
- nppsAddC\_32sc\_ISfs
  - signal\_addc, [2252](#)
- nppsAddC\_32sc\_Sfs
  - signal\_addc, [2252](#)
- nppsAddC\_64f
  - signal\_addc, [2253](#)
- nppsAddC\_64f\_I
  - signal\_addc, [2253](#)
- nppsAddC\_64fc
  - signal\_addc, [2253](#)
- nppsAddC\_64fc\_I
  - signal\_addc, [2254](#)
- nppsAddC\_8u\_ISfs
  - signal\_addc, [2254](#)
- nppsAddC\_8u\_Sfs
  - signal\_addc, [2254](#)
- nppsAddProduct\_16s32s\_Sfs
  - signal\_addproduct, [2307](#)
- nppsAddProduct\_16s\_Sfs
  - signal\_addproduct, [2307](#)
- nppsAddProduct\_32f
  - signal\_addproduct, [2307](#)
- nppsAddProduct\_32fc
  - signal\_addproduct, [2308](#)
- nppsAddProduct\_32s\_Sfs
  - signal\_addproduct, [2308](#)
- nppsAddProduct\_64f
  - signal\_addproduct, [2308](#)
- nppsAddProduct\_64fc
  - signal\_addproduct, [2309](#)
- nppsAddProductC\_32f
  - signal\_addproductc, [2256](#)
- nppsAnd\_16u
  - signal\_and, [2386](#)
- nppsAnd\_16u\_I
  - signal\_and, [2386](#)
- nppsAnd\_32u
  - signal\_and, [2387](#)
- nppsAnd\_32u\_I
  - signal\_and, [2387](#)
- nppsAnd\_8u
  - signal\_and, [2387](#)
- nppsAnd\_8u\_I
  - signal\_and, [2388](#)
- nppsAndC\_16u
  - signal\_andc, [2383](#)
- nppsAndC\_16u\_I
  - signal\_andc, [2383](#)
- nppsAndC\_32u
  - signal\_andc, [2384](#)
- nppsAndC\_32u\_I
  - signal\_andc, [2384](#)
- nppsAndC\_8u
  - signal\_andc, [2384](#)
- nppsAndC\_8u\_I
  - signal\_andc, [2385](#)
- nppsArctan\_32f
  - signal\_inversetan, [2375](#)
- nppsArctan\_32f\_I
  - signal\_inversetan, [2375](#)
- nppsArctan\_64f
  - signal\_inversetan, [2375](#)
- nppsArctan\_64f\_I
  - signal\_inversetan, [2376](#)
- nppsAverageError\_16s
  - signal\_average\_error, [2584](#)
- nppsAverageError\_16sc
  - signal\_average\_error, [2584](#)
- nppsAverageError\_16u
  - signal\_average\_error, [2584](#)
- nppsAverageError\_32f
  - signal\_average\_error, [2585](#)
- nppsAverageError\_32fc
  - signal\_average\_error, [2585](#)
- nppsAverageError\_32s
  - signal\_average\_error, [2585](#)
- nppsAverageError\_32sc
  - signal\_average\_error, [2586](#)
- nppsAverageError\_32u
  - signal\_average\_error, [2586](#)
- nppsAverageError\_64f
  - signal\_average\_error, [2586](#)
- nppsAverageError\_64fc
  - signal\_average\_error, [2587](#)
- nppsAverageError\_64s
  - signal\_average\_error, [2587](#)
- nppsAverageError\_64sc
  - signal\_average\_error, [2587](#)
- nppsAverageError\_8s
  - signal\_average\_error, [2588](#)
- nppsAverageError\_8u
  - signal\_average\_error, [2588](#)
- nppsAverageErrorGetBufferSize\_16s
  - signal\_average\_error, [2588](#)
- nppsAverageErrorGetBufferSize\_16sc
  - signal\_average\_error, [2589](#)
- nppsAverageErrorGetBufferSize\_16u
  - signal\_average\_error, [2589](#)
- nppsAverageErrorGetBufferSize\_32f
  - signal\_average\_error, [2589](#)
- nppsAverageErrorGetBufferSize\_32fc
  - signal\_average\_error, [2589](#)

- npplsAverageErrorGetBufferSize\_32s  
  signal\_average\_error, [2590](#)
- npplsAverageErrorGetBufferSize\_32sc  
  signal\_average\_error, [2590](#)
- npplsAverageErrorGetBufferSize\_32u  
  signal\_average\_error, [2590](#)
- npplsAverageErrorGetBufferSize\_64f  
  signal\_average\_error, [2590](#)
- npplsAverageErrorGetBufferSize\_64fc  
  signal\_average\_error, [2591](#)
- npplsAverageErrorGetBufferSize\_64s  
  signal\_average\_error, [2591](#)
- npplsAverageErrorGetBufferSize\_64sc  
  signal\_average\_error, [2591](#)
- npplsAverageErrorGetBufferSize\_8s  
  signal\_average\_error, [2591](#)
- npplsAverageErrorGetBufferSize\_8u  
  signal\_average\_error, [2592](#)
- npplsAverageRelativeError\_16s  
  signal\_average\_relative\_error, [2607](#)
- npplsAverageRelativeError\_16sc  
  signal\_average\_relative\_error, [2607](#)
- npplsAverageRelativeError\_16u  
  signal\_average\_relative\_error, [2608](#)
- npplsAverageRelativeError\_32f  
  signal\_average\_relative\_error, [2608](#)
- npplsAverageRelativeError\_32fc  
  signal\_average\_relative\_error, [2608](#)
- npplsAverageRelativeError\_32s  
  signal\_average\_relative\_error, [2609](#)
- npplsAverageRelativeError\_32sc  
  signal\_average\_relative\_error, [2609](#)
- npplsAverageRelativeError\_32u  
  signal\_average\_relative\_error, [2610](#)
- npplsAverageRelativeError\_64f  
  signal\_average\_relative\_error, [2610](#)
- npplsAverageRelativeError\_64fc  
  signal\_average\_relative\_error, [2610](#)
- npplsAverageRelativeError\_64s  
  signal\_average\_relative\_error, [2611](#)
- npplsAverageRelativeError\_64sc  
  signal\_average\_relative\_error, [2611](#)
- npplsAverageRelativeError\_8s  
  signal\_average\_relative\_error, [2612](#)
- npplsAverageRelativeError\_8u  
  signal\_average\_relative\_error, [2612](#)
- npplsAverageRelativeErrorGetBufferSize\_16s  
  signal\_average\_relative\_error, [2612](#)
- npplsAverageRelativeErrorGetBufferSize\_16sc  
  signal\_average\_relative\_error, [2613](#)
- npplsAverageRelativeErrorGetBufferSize\_16u  
  signal\_average\_relative\_error, [2613](#)
- npplsAverageRelativeErrorGetBufferSize\_32f  
  signal\_average\_relative\_error, [2613](#)
- npplsAverageRelativeErrorGetBufferSize\_32fc  
  signal\_average\_relative\_error, [2613](#)
- npplsAverageRelativeErrorGetBufferSize\_32s  
  signal\_average\_relative\_error, [2614](#)
- npplsAverageRelativeErrorGetBufferSize\_32sc  
  signal\_average\_relative\_error, [2614](#)
- npplsAverageRelativeErrorGetBufferSize\_32u  
  signal\_average\_relative\_error, [2614](#)
- npplsAverageRelativeErrorGetBufferSize\_64f  
  signal\_average\_relative\_error, [2614](#)
- npplsAverageRelativeErrorGetBufferSize\_64fc  
  signal\_average\_relative\_error, [2615](#)
- npplsAverageRelativeErrorGetBufferSize\_64s  
  signal\_average\_relative\_error, [2615](#)
- npplsAverageRelativeErrorGetBufferSize\_64sc  
  signal\_average\_relative\_error, [2615](#)
- npplsAverageRelativeErrorGetBufferSize\_8s  
  signal\_average\_relative\_error, [2615](#)
- npplsAverageRelativeErrorGetBufferSize\_8u  
  signal\_average\_relative\_error, [2616](#)
- npplsCauchy\_32f\_I  
  signal\_cauchy, [2380](#)
- npplsCauchyD\_32f\_I  
  signal\_cauchy, [2380](#)
- npplsCauchyDD2\_32f\_I  
  signal\_cauchy, [2380](#)
- npplsConvert\_16s32f  
  signal\_convert, [2415](#)
- npplsConvert\_16s32f\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_16s32s  
  signal\_convert, [2415](#)
- npplsConvert\_16s64f\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_16s8s\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_16u32f  
  signal\_convert, [2415](#)
- npplsConvert\_32f16s\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_32f16u\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_32f32s\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_32f64f  
  signal\_convert, [2415](#)
- npplsConvert\_32f8s\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_32f8u\_Sfs  
  signal\_convert, [2415](#)
- npplsConvert\_32s16s  
  signal\_convert, [2415](#)
- npplsConvert\_32s16s\_Sfs  
  signal\_convert, [2415](#)



- nppsConvert\_32s32f
  - signal\_convert, [2415](#)
- nppsConvert\_32s32f\_Sfs
  - signal\_convert, [2415](#)
- nppsConvert\_32s64f
  - signal\_convert, [2415](#)
- nppsConvert\_32s64f\_Sfs
  - signal\_convert, [2415](#)
- nppsConvert\_64f16s\_Sfs
  - signal\_convert, [2415](#)
- nppsConvert\_64f32f
  - signal\_convert, [2415](#)
- nppsConvert\_64f32s\_Sfs
  - signal\_convert, [2415](#)
- nppsConvert\_64f64s\_Sfs
  - signal\_convert, [2415](#)
- nppsConvert\_64s32s\_Sfs
  - signal\_convert, [2415](#)
- nppsConvert\_64s64f
  - signal\_convert, [2415](#)
- nppsConvert\_8s16s
  - signal\_convert, [2415](#)
- nppsConvert\_8s32f
  - signal\_convert, [2415](#)
- nppsConvert\_8u32f
  - signal\_convert, [2415](#)
- nppsCopy\_16s
  - signal\_copy, [2453](#)
- nppsCopy\_16sc
  - signal\_copy, [2454](#)
- nppsCopy\_32f
  - signal\_copy, [2454](#)
- nppsCopy\_32fc
  - signal\_copy, [2454](#)
- nppsCopy\_32s
  - signal\_copy, [2454](#)
- nppsCopy\_32sc
  - signal\_copy, [2455](#)
- nppsCopy\_64fc
  - signal\_copy, [2455](#)
- nppsCopy\_64s
  - signal\_copy, [2455](#)
- nppsCopy\_64sc
  - signal\_copy, [2456](#)
- nppsCopy\_8u
  - signal\_copy, [2456](#)
- nppsCountInRange\_32s
  - signal\_count\_in\_range, [2568](#)
- nppsCountInRangeGetBufferSize\_32s
  - signal\_count\_in\_range, [2568](#)
- nppsCubrt\_32f
  - signal\_cuberoot, [2361](#)
- nppsCubrt\_32s16s\_Sfs
  - signal\_cuberoot, [2361](#)
- nppsDiv\_16s\_ISfs
  - signal\_div, [2334](#)
- nppsDiv\_16s\_Sfs
  - signal\_div, [2334](#)
- nppsDiv\_16sc\_ISfs
  - signal\_div, [2335](#)
- nppsDiv\_16sc\_Sfs
  - signal\_div, [2335](#)
- nppsDiv\_16u\_ISfs
  - signal\_div, [2335](#)
- nppsDiv\_16u\_Sfs
  - signal\_div, [2336](#)
- nppsDiv\_32f
  - signal\_div, [2336](#)
- nppsDiv\_32f\_I
  - signal\_div, [2336](#)
- nppsDiv\_32fc
  - signal\_div, [2337](#)
- nppsDiv\_32fc\_I
  - signal\_div, [2337](#)
- nppsDiv\_32s16s\_Sfs
  - signal\_div, [2337](#)
- nppsDiv\_32s\_ISfs
  - signal\_div, [2338](#)
- nppsDiv\_32s\_Sfs
  - signal\_div, [2338](#)
- nppsDiv\_64f
  - signal\_div, [2338](#)
- nppsDiv\_64f\_I
  - signal\_div, [2339](#)
- nppsDiv\_64fc
  - signal\_div, [2339](#)
- nppsDiv\_64fc\_I
  - signal\_div, [2339](#)
- nppsDiv\_8u\_ISfs
  - signal\_div, [2340](#)
- nppsDiv\_8u\_Sfs
  - signal\_div, [2340](#)
- nppsDiv\_Round\_16s\_ISfs
  - signal\_divround, [2341](#)
- nppsDiv\_Round\_16s\_Sfs
  - signal\_divround, [2342](#)
- nppsDiv\_Round\_16u\_ISfs
  - signal\_divround, [2342](#)
- nppsDiv\_Round\_16u\_Sfs
  - signal\_divround, [2342](#)
- nppsDiv\_Round\_8u\_ISfs
  - signal\_divround, [2343](#)
- nppsDiv\_Round\_8u\_Sfs
  - signal\_divround, [2343](#)
- nppsDivC\_16s\_ISfs
  - signal\_divc, [2286](#)
- nppsDivC\_16s\_Sfs
  - signal\_divc, [2286](#)

- signal\_divc, [2286](#)
- nppsDivC\_16sc\_Sfs
  - signal\_divc, [2287](#)
- nppsDivC\_16u\_ISfs
  - signal\_divc, [2287](#)
- nppsDivC\_16u\_Sfs
  - signal\_divc, [2287](#)
- nppsDivC\_32f
  - signal\_divc, [2288](#)
- nppsDivC\_32f\_I
  - signal\_divc, [2288](#)
- nppsDivC\_32fc
  - signal\_divc, [2288](#)
- nppsDivC\_32fc\_I
  - signal\_divc, [2289](#)
- nppsDivC\_64f
  - signal\_divc, [2289](#)
- nppsDivC\_64f\_I
  - signal\_divc, [2289](#)
- nppsDivC\_64fc
  - signal\_divc, [2290](#)
- nppsDivC\_64fc\_I
  - signal\_divc, [2290](#)
- nppsDivC\_8u\_ISfs
  - signal\_divc, [2290](#)
- nppsDivC\_8u\_Sfs
  - signal\_divc, [2291](#)
- nppsDivCRev\_16u
  - signal\_divcrev, [2292](#)
- nppsDivCRev\_16u\_I
  - signal\_divcrev, [2292](#)
- nppsDivCRev\_32f
  - signal\_divcrev, [2293](#)
- nppsDivCRev\_32f\_I
  - signal\_divcrev, [2293](#)
- nppsDotProd\_16s16sc32fc
  - signal\_dot\_product, [2551](#)
- nppsDotProd\_16s16sc32sc\_Sfs
  - signal\_dot\_product, [2552](#)
- nppsDotProd\_16s16sc64sc
  - signal\_dot\_product, [2552](#)
- nppsDotProd\_16s16sc\_Sfs
  - signal\_dot\_product, [2552](#)
- nppsDotProd\_16s32f
  - signal\_dot\_product, [2553](#)
- nppsDotProd\_16s32s32s\_Sfs
  - signal\_dot\_product, [2553](#)
- nppsDotProd\_16s32s\_Sfs
  - signal\_dot\_product, [2554](#)
- nppsDotProd\_16s64s
  - signal\_dot\_product, [2554](#)
- nppsDotProd\_16s\_Sfs
  - signal\_dot\_product, [2554](#)
- nppsDotProd\_16sc32fc
  - signal\_dot\_product, [2555](#)
- nppsDotProd\_16sc32sc\_Sfs
  - signal\_dot\_product, [2555](#)
- nppsDotProd\_16sc64sc
  - signal\_dot\_product, [2556](#)
- nppsDotProd\_16sc\_Sfs
  - signal\_dot\_product, [2556](#)
- nppsDotProd\_32f
  - signal\_dot\_product, [2556](#)
- nppsDotProd\_32f32fc
  - signal\_dot\_product, [2557](#)
- nppsDotProd\_32f32fc64fc
  - signal\_dot\_product, [2557](#)
- nppsDotProd\_32f64f
  - signal\_dot\_product, [2557](#)
- nppsDotProd\_32fc
  - signal\_dot\_product, [2558](#)
- nppsDotProd\_32fc64fc
  - signal\_dot\_product, [2558](#)
- nppsDotProd\_32s32sc\_Sfs
  - signal\_dot\_product, [2558](#)
- nppsDotProd\_32s\_Sfs
  - signal\_dot\_product, [2559](#)
- nppsDotProd\_32sc\_Sfs
  - signal\_dot\_product, [2559](#)
- nppsDotProd\_64f
  - signal\_dot\_product, [2560](#)
- nppsDotProd\_64f64fc
  - signal\_dot\_product, [2560](#)
- nppsDotProd\_64fc
  - signal\_dot\_product, [2560](#)
- nppsDotProdGetBufferSize\_16s16sc32fc
  - signal\_dot\_product, [2561](#)
- nppsDotProdGetBufferSize\_16s16sc32sc\_Sfs
  - signal\_dot\_product, [2561](#)
- nppsDotProdGetBufferSize\_16s16sc64sc
  - signal\_dot\_product, [2561](#)
- nppsDotProdGetBufferSize\_16s16sc\_Sfs
  - signal\_dot\_product, [2562](#)
- nppsDotProdGetBufferSize\_16s32f
  - signal\_dot\_product, [2562](#)
- nppsDotProdGetBufferSize\_16s32s32s\_Sfs
  - signal\_dot\_product, [2562](#)
- nppsDotProdGetBufferSize\_16s32s\_Sfs
  - signal\_dot\_product, [2562](#)
- nppsDotProdGetBufferSize\_16s64s
  - signal\_dot\_product, [2563](#)
- nppsDotProdGetBufferSize\_16s\_Sfs
  - signal\_dot\_product, [2563](#)
- nppsDotProdGetBufferSize\_16sc32fc
  - signal\_dot\_product, [2563](#)
- nppsDotProdGetBufferSize\_16sc32sc\_Sfs
  - signal\_dot\_product, [2563](#)

- nppsDotProdGetBufferSize\_16sc64sc
  - signal\_dot\_product, [2564](#)
- nppsDotProdGetBufferSize\_16sc\_Sfs
  - signal\_dot\_product, [2564](#)
- nppsDotProdGetBufferSize\_32f
  - signal\_dot\_product, [2564](#)
- nppsDotProdGetBufferSize\_32f32fc
  - signal\_dot\_product, [2564](#)
- nppsDotProdGetBufferSize\_32f32fc64fc
  - signal\_dot\_product, [2565](#)
- nppsDotProdGetBufferSize\_32f64f
  - signal\_dot\_product, [2565](#)
- nppsDotProdGetBufferSize\_32fc
  - signal\_dot\_product, [2565](#)
- nppsDotProdGetBufferSize\_32fc64fc
  - signal\_dot\_product, [2565](#)
- nppsDotProdGetBufferSize\_32s32sc\_Sfs
  - signal\_dot\_product, [2566](#)
- nppsDotProdGetBufferSize\_32s\_Sfs
  - signal\_dot\_product, [2566](#)
- nppsDotProdGetBufferSize\_32sc\_Sfs
  - signal\_dot\_product, [2566](#)
- nppsDotProdGetBufferSize\_64f
  - signal\_dot\_product, [2566](#)
- nppsDotProdGetBufferSize\_64f64fc
  - signal\_dot\_product, [2567](#)
- nppsDotProdGetBufferSize\_64fc
  - signal\_dot\_product, [2567](#)
- nppSetStream
  - core\_npp, [33](#)
- nppsExp\_16s\_ISfs
  - signal\_exp, [2362](#)
- nppsExp\_16s\_Sfs
  - signal\_exp, [2363](#)
- nppsExp\_32f
  - signal\_exp, [2363](#)
- nppsExp\_32f64f
  - signal\_exp, [2363](#)
- nppsExp\_32f\_I
  - signal\_exp, [2363](#)
- nppsExp\_32s\_ISfs
  - signal\_exp, [2364](#)
- nppsExp\_32s\_Sfs
  - signal\_exp, [2364](#)
- nppsExp\_64f
  - signal\_exp, [2364](#)
- nppsExp\_64f\_I
  - signal\_exp, [2365](#)
- nppsExp\_64s\_ISfs
  - signal\_exp, [2365](#)
- nppsExp\_64s\_Sfs
  - signal\_exp, [2365](#)
- nppsFree
  - signal\_free, [2623](#)
- nppsIntegral\_32s
  - signal\_integral, [2442](#)
- nppsIntegralGetBufferSize\_32s
  - signal\_integral, [2442](#)
- nppsLn\_16s\_ISfs
  - signal\_ln, [2366](#)
- nppsLn\_16s\_Sfs
  - signal\_ln, [2367](#)
- nppsLn\_32f
  - signal\_ln, [2367](#)
- nppsLn\_32f\_I
  - signal\_ln, [2367](#)
- nppsLn\_32s16s\_Sfs
  - signal\_ln, [2367](#)
- nppsLn\_32s\_ISfs
  - signal\_ln, [2368](#)
- nppsLn\_32s\_Sfs
  - signal\_ln, [2368](#)
- nppsLn\_64f
  - signal\_ln, [2368](#)
- nppsLn\_64f32f
  - signal\_ln, [2369](#)
- nppsLn\_64f\_I
  - signal\_ln, [2369](#)
- nppsLShiftC\_16s
  - signal\_lshiftc, [2404](#)
- nppsLShiftC\_16s\_I
  - signal\_lshiftc, [2405](#)
- nppsLShiftC\_16u
  - signal\_lshiftc, [2405](#)
- nppsLShiftC\_16u\_I
  - signal\_lshiftc, [2405](#)
- nppsLShiftC\_32s
  - signal\_lshiftc, [2405](#)
- nppsLShiftC\_32s\_I
  - signal\_lshiftc, [2406](#)
- nppsLShiftC\_32u
  - signal\_lshiftc, [2406](#)
- nppsLShiftC\_32u\_I
  - signal\_lshiftc, [2406](#)
- nppsLShiftC\_8u
  - signal\_lshiftc, [2407](#)
- nppsLShiftC\_8u\_I
  - signal\_lshiftc, [2407](#)
- nppsMalloc\_16s
  - signal\_malloc, [2619](#)
- nppsMalloc\_16sc
  - signal\_malloc, [2619](#)
- nppsMalloc\_16u
  - signal\_malloc, [2619](#)
- nppsMalloc\_32f
  - signal\_malloc, [2619](#)
- nppsMalloc\_32fc
  - signal\_malloc, [2620](#)

- npplsMalloc\_32s
  - signal\_malloc, [2620](#)
- npplsMalloc\_32sc
  - signal\_malloc, [2620](#)
- npplsMalloc\_32u
  - signal\_malloc, [2620](#)
- npplsMalloc\_64f
  - signal\_malloc, [2621](#)
- npplsMalloc\_64fc
  - signal\_malloc, [2621](#)
- npplsMalloc\_64s
  - signal\_malloc, [2621](#)
- npplsMalloc\_64sc
  - signal\_malloc, [2621](#)
- npplsMalloc\_8s
  - signal\_malloc, [2622](#)
- npplsMalloc\_8u
  - signal\_malloc, [2622](#)
- npplsMax\_16s
  - signal\_max, [2470](#)
- npplsMax\_32f
  - signal\_max, [2471](#)
- npplsMax\_32s
  - signal\_max, [2471](#)
- npplsMax\_64f
  - signal\_max, [2471](#)
- npplsMaxAbs\_16s
  - signal\_max, [2472](#)
- npplsMaxAbs\_32s
  - signal\_max, [2472](#)
- npplsMaxAbsGetBufferSize\_16s
  - signal\_max, [2472](#)
- npplsMaxAbsGetBufferSize\_32s
  - signal\_max, [2473](#)
- npplsMaxAbsIndx\_16s
  - signal\_max, [2473](#)
- npplsMaxAbsIndx\_32s
  - signal\_max, [2473](#)
- npplsMaxAbsIndxGetBufferSize\_16s
  - signal\_max, [2474](#)
- npplsMaxAbsIndxGetBufferSize\_32s
  - signal\_max, [2474](#)
- npplsMaxEvery\_16s\_I
  - signal\_min\_every\_or\_max\_every, [2458](#)
- npplsMaxEvery\_16u\_I
  - signal\_min\_every\_or\_max\_every, [2459](#)
- npplsMaxEvery\_32f\_I
  - signal\_min\_every\_or\_max\_every, [2459](#)
- npplsMaxEvery\_32s\_I
  - signal\_min\_every\_or\_max\_every, [2459](#)
- npplsMaxEvery\_8u\_I
  - signal\_min\_every\_or\_max\_every, [2459](#)
- npplsMaxGetBufferSize\_16s
  - signal\_max, [2474](#)
- npplsMaxGetBufferSize\_32f
  - signal\_max, [2474](#)
- npplsMaxGetBufferSize\_32s
  - signal\_max, [2475](#)
- npplsMaxGetBufferSize\_64f
  - signal\_max, [2475](#)
- npplsMaximumError\_16s
  - signal\_maximum\_error, [2573](#)
- npplsMaximumError\_16sc
  - signal\_maximum\_error, [2573](#)
- npplsMaximumError\_16u
  - signal\_maximum\_error, [2573](#)
- npplsMaximumError\_32f
  - signal\_maximum\_error, [2574](#)
- npplsMaximumError\_32fc
  - signal\_maximum\_error, [2574](#)
- npplsMaximumError\_32s
  - signal\_maximum\_error, [2574](#)
- npplsMaximumError\_32sc
  - signal\_maximum\_error, [2575](#)
- npplsMaximumError\_32u
  - signal\_maximum\_error, [2575](#)
- npplsMaximumError\_64f
  - signal\_maximum\_error, [2575](#)
- npplsMaximumError\_64fc
  - signal\_maximum\_error, [2576](#)
- npplsMaximumError\_64s
  - signal\_maximum\_error, [2576](#)
- npplsMaximumError\_64sc
  - signal\_maximum\_error, [2576](#)
- npplsMaximumError\_8s
  - signal\_maximum\_error, [2577](#)
- npplsMaximumError\_8u
  - signal\_maximum\_error, [2577](#)
- npplsMaximumErrorGetBufferSize\_16s
  - signal\_maximum\_error, [2577](#)
- npplsMaximumErrorGetBufferSize\_16sc
  - signal\_maximum\_error, [2578](#)
- npplsMaximumErrorGetBufferSize\_16u
  - signal\_maximum\_error, [2578](#)
- npplsMaximumErrorGetBufferSize\_32f
  - signal\_maximum\_error, [2578](#)
- npplsMaximumErrorGetBufferSize\_32fc
  - signal\_maximum\_error, [2578](#)
- npplsMaximumErrorGetBufferSize\_32s
  - signal\_maximum\_error, [2579](#)
- npplsMaximumErrorGetBufferSize\_32sc
  - signal\_maximum\_error, [2579](#)
- npplsMaximumErrorGetBufferSize\_32u
  - signal\_maximum\_error, [2579](#)
- npplsMaximumErrorGetBufferSize\_64f
  - signal\_maximum\_error, [2579](#)
- npplsMaximumErrorGetBufferSize\_64fc
  - signal\_maximum\_error, [2580](#)

- nppsMaximumErrorGetBufferSize\_64s
  - signal\_maximum\_error, [2580](#)
- nppsMaximumErrorGetBufferSize\_64sc
  - signal\_maximum\_error, [2580](#)
- nppsMaximumErrorGetBufferSize\_8s
  - signal\_maximum\_error, [2580](#)
- nppsMaximumErrorGetBufferSize\_8u
  - signal\_maximum\_error, [2581](#)
- nppsMaximumRelativeError\_16s
  - signal\_maximum\_relative\_error, [2595](#)
- nppsMaximumRelativeError\_16sc
  - signal\_maximum\_relative\_error, [2595](#)
- nppsMaximumRelativeError\_16u
  - signal\_maximum\_relative\_error, [2596](#)
- nppsMaximumRelativeError\_32f
  - signal\_maximum\_relative\_error, [2596](#)
- nppsMaximumRelativeError\_32fc
  - signal\_maximum\_relative\_error, [2596](#)
- nppsMaximumRelativeError\_32s
  - signal\_maximum\_relative\_error, [2597](#)
- nppsMaximumRelativeError\_32sc
  - signal\_maximum\_relative\_error, [2597](#)
- nppsMaximumRelativeError\_32u
  - signal\_maximum\_relative\_error, [2598](#)
- nppsMaximumRelativeError\_64f
  - signal\_maximum\_relative\_error, [2598](#)
- nppsMaximumRelativeError\_64fc
  - signal\_maximum\_relative\_error, [2598](#)
- nppsMaximumRelativeError\_64s
  - signal\_maximum\_relative\_error, [2599](#)
- nppsMaximumRelativeError\_64sc
  - signal\_maximum\_relative\_error, [2599](#)
- nppsMaximumRelativeError\_8s
  - signal\_maximum\_relative\_error, [2600](#)
- nppsMaximumRelativeError\_8u
  - signal\_maximum\_relative\_error, [2600](#)
- nppsMaximumRelativeErrorGetBufferSize\_16s
  - signal\_maximum\_relative\_error, [2600](#)
- nppsMaximumRelativeErrorGetBufferSize\_16sc
  - signal\_maximum\_relative\_error, [2601](#)
- nppsMaximumRelativeErrorGetBufferSize\_16u
  - signal\_maximum\_relative\_error, [2601](#)
- nppsMaximumRelativeErrorGetBufferSize\_32f
  - signal\_maximum\_relative\_error, [2601](#)
- nppsMaximumRelativeErrorGetBufferSize\_32fc
  - signal\_maximum\_relative\_error, [2601](#)
- nppsMaximumRelativeErrorGetBufferSize\_32s
  - signal\_maximum\_relative\_error, [2602](#)
- nppsMaximumRelativeErrorGetBufferSize\_32sc
  - signal\_maximum\_relative\_error, [2602](#)
- nppsMaximumRelativeErrorGetBufferSize\_32u
  - signal\_maximum\_relative\_error, [2602](#)
- nppsMaximumRelativeErrorGetBufferSize\_64f
  - signal\_maximum\_relative\_error, [2602](#)
- nppsMaximumRelativeErrorGetBufferSize\_64fc
  - signal\_maximum\_relative\_error, [2603](#)
- nppsMaximumRelativeErrorGetBufferSize\_64s
  - signal\_maximum\_relative\_error, [2603](#)
- nppsMaximumRelativeErrorGetBufferSize\_64sc
  - signal\_maximum\_relative\_error, [2603](#)
- nppsMaximumRelativeErrorGetBufferSize\_8s
  - signal\_maximum\_relative\_error, [2603](#)
- nppsMaximumRelativeErrorGetBufferSize\_8u
  - signal\_maximum\_relative\_error, [2604](#)
- nppsMaxIndx\_16s
  - signal\_max, [2475](#)
- nppsMaxIndx\_32f
  - signal\_max, [2476](#)
- nppsMaxIndx\_32s
  - signal\_max, [2476](#)
- nppsMaxIndx\_64f
  - signal\_max, [2476](#)
- nppsMaxIndxGetBufferSize\_16s
  - signal\_max, [2477](#)
- nppsMaxIndxGetBufferSize\_32f
  - signal\_max, [2477](#)
- nppsMaxIndxGetBufferSize\_32s
  - signal\_max, [2477](#)
- nppsMaxIndxGetBufferSize\_64f
  - signal\_max, [2478](#)
- nppsMean\_16s\_Sfs
  - signal\_mean, [2490](#)
- nppsMean\_16sc\_Sfs
  - signal\_mean, [2490](#)
- nppsMean\_32f
  - signal\_mean, [2490](#)
- nppsMean\_32fc
  - signal\_mean, [2491](#)
- nppsMean\_32s\_Sfs
  - signal\_mean, [2491](#)
- nppsMean\_64f
  - signal\_mean, [2491](#)
- nppsMean\_64fc
  - signal\_mean, [2492](#)
- nppsMeanGetBufferSize\_16s\_Sfs
  - signal\_mean, [2492](#)
- nppsMeanGetBufferSize\_16sc\_Sfs
  - signal\_mean, [2492](#)
- nppsMeanGetBufferSize\_32f
  - signal\_mean, [2493](#)
- nppsMeanGetBufferSize\_32fc
  - signal\_mean, [2493](#)
- nppsMeanGetBufferSize\_32s\_Sfs
  - signal\_mean, [2493](#)
- nppsMeanGetBufferSize\_64f
  - signal\_mean, [2493](#)
- nppsMeanGetBufferSize\_64fc
  - signal\_mean, [2494](#)

- nppsMeanStdDev\_16s32s\_Sfs
  - signal\_mean\_and\_standard\_deviation, 2498
- nppsMeanStdDev\_16s\_Sfs
  - signal\_mean\_and\_standard\_deviation, 2499
- nppsMeanStdDev\_32f
  - signal\_mean\_and\_standard\_deviation, 2499
- nppsMeanStdDev\_64f
  - signal\_mean\_and\_standard\_deviation, 2499
- nppsMeanStdDevGetBufferSize\_16s32s\_Sfs
  - signal\_mean\_and\_standard\_deviation, 2500
- nppsMeanStdDevGetBufferSize\_16s\_Sfs
  - signal\_mean\_and\_standard\_deviation, 2500
- nppsMeanStdDevGetBufferSize\_32f
  - signal\_mean\_and\_standard\_deviation, 2500
- nppsMeanStdDevGetBufferSize\_64f
  - signal\_mean\_and\_standard\_deviation, 2500
- nppsMin\_16s
  - signal\_min, 2480
- nppsMin\_32f
  - signal\_min, 2481
- nppsMin\_32s
  - signal\_min, 2481
- nppsMin\_64f
  - signal\_min, 2481
- nppsMinAbs\_16s
  - signal\_min, 2482
- nppsMinAbs\_32s
  - signal\_min, 2482
- nppsMinAbsGetBufferSize\_16s
  - signal\_min, 2482
- nppsMinAbsGetBufferSize\_32s
  - signal\_min, 2483
- nppsMinAbsIdx\_16s
  - signal\_min, 2483
- nppsMinAbsIdx\_32s
  - signal\_min, 2483
- nppsMinAbsIdxGetBufferSize\_16s
  - signal\_min, 2484
- nppsMinAbsIdxGetBufferSize\_32s
  - signal\_min, 2484
- nppsMinEvery\_16s\_I
  - signal\_min\_every\_or\_max\_every, 2460
- nppsMinEvery\_16u\_I
  - signal\_min\_every\_or\_max\_every, 2460
- nppsMinEvery\_32f\_I
  - signal\_min\_every\_or\_max\_every, 2460
- nppsMinEvery\_32s\_I
  - signal\_min\_every\_or\_max\_every, 2461
- nppsMinEvery\_64f\_I
  - signal\_min\_every\_or\_max\_every, 2461
- nppsMinEvery\_8u\_I
  - signal\_min\_every\_or\_max\_every, 2461
- nppsMinGetBufferSize\_16s
  - signal\_min, 2484
- nppsMinGetBufferSize\_32f
  - signal\_min, 2484
- nppsMinGetBufferSize\_32s
  - signal\_min, 2485
- nppsMinGetBufferSize\_64f
  - signal\_min, 2485
- nppsMinIdx\_16s
  - signal\_min, 2485
- nppsMinIdx\_32f
  - signal\_min, 2486
- nppsMinIdx\_32s
  - signal\_min, 2486
- nppsMinIdx\_64f
  - signal\_min, 2486
- nppsMinIdxGetBufferSize\_16s
  - signal\_min, 2487
- nppsMinIdxGetBufferSize\_32f
  - signal\_min, 2487
- nppsMinIdxGetBufferSize\_32s
  - signal\_min, 2487
- nppsMinIdxGetBufferSize\_64f
  - signal\_min, 2488
- nppsMinMax\_16s
  - signal\_min\_max, 2504
- nppsMinMax\_16u
  - signal\_min\_max, 2504
- nppsMinMax\_32f
  - signal\_min\_max, 2504
- nppsMinMax\_32s
  - signal\_min\_max, 2505
- nppsMinMax\_32u
  - signal\_min\_max, 2505
- nppsMinMax\_64f
  - signal\_min\_max, 2505
- nppsMinMax\_8u
  - signal\_min\_max, 2506
- nppsMinMaxGetBufferSize\_16s
  - signal\_min\_max, 2506
- nppsMinMaxGetBufferSize\_16u
  - signal\_min\_max, 2506
- nppsMinMaxGetBufferSize\_32f
  - signal\_min\_max, 2507
- nppsMinMaxGetBufferSize\_32s
  - signal\_min\_max, 2507
- nppsMinMaxGetBufferSize\_32u
  - signal\_min\_max, 2507
- nppsMinMaxGetBufferSize\_64f
  - signal\_min\_max, 2507
- nppsMinMaxGetBufferSize\_8u
  - signal\_min\_max, 2508
- nppsMinMaxIdx\_16s
  - signal\_min\_max, 2508
- nppsMinMaxIdx\_16u
  - signal\_min\_max, 2508



- nppsMinMaxIndx\_32f
  - signal\_min\_max, [2509](#)
- nppsMinMaxIndx\_32s
  - signal\_min\_max, [2509](#)
- nppsMinMaxIndx\_32u
  - signal\_min\_max, [2510](#)
- nppsMinMaxIndx\_64f
  - signal\_min\_max, [2510](#)
- nppsMinMaxIndx\_8u
  - signal\_min\_max, [2510](#)
- nppsMinMaxIndxGetBufferSize\_16s
  - signal\_min\_max, [2511](#)
- nppsMinMaxIndxGetBufferSize\_16u
  - signal\_min\_max, [2511](#)
- nppsMinMaxIndxGetBufferSize\_32f
  - signal\_min\_max, [2511](#)
- nppsMinMaxIndxGetBufferSize\_32s
  - signal\_min\_max, [2512](#)
- nppsMinMaxIndxGetBufferSize\_32u
  - signal\_min\_max, [2512](#)
- nppsMinMaxIndxGetBufferSize\_64f
  - signal\_min\_max, [2512](#)
- nppsMinMaxIndxGetBufferSize\_8u
  - signal\_min\_max, [2512](#)
- nppsMul\_16s
  - signal\_mul, [2312](#)
- nppsMul\_16s32f
  - signal\_mul, [2312](#)
- nppsMul\_16s32s\_Sfs
  - signal\_mul, [2313](#)
- nppsMul\_16s\_I
  - signal\_mul, [2313](#)
- nppsMul\_16s\_ISfs
  - signal\_mul, [2313](#)
- nppsMul\_16s\_Sfs
  - signal\_mul, [2314](#)
- nppsMul\_16sc\_ISfs
  - signal\_mul, [2314](#)
- nppsMul\_16sc\_Sfs
  - signal\_mul, [2314](#)
- nppsMul\_16u16s\_Sfs
  - signal\_mul, [2315](#)
- nppsMul\_16u\_ISfs
  - signal\_mul, [2315](#)
- nppsMul\_16u\_Sfs
  - signal\_mul, [2315](#)
- nppsMul\_32f
  - signal\_mul, [2316](#)
- nppsMul\_32f32fc
  - signal\_mul, [2316](#)
- nppsMul\_32f32fc\_I
  - signal\_mul, [2316](#)
- nppsMul\_32f\_I
  - signal\_mul, [2317](#)
- nppsMul\_32fc
  - signal\_mul, [2317](#)
- nppsMul\_32fc\_I
  - signal\_mul, [2317](#)
- nppsMul\_32s32sc\_ISfs
  - signal\_mul, [2318](#)
- nppsMul\_32s32sc\_Sfs
  - signal\_mul, [2318](#)
- nppsMul\_32s\_ISfs
  - signal\_mul, [2318](#)
- nppsMul\_32s\_Sfs
  - signal\_mul, [2319](#)
- nppsMul\_32sc\_ISfs
  - signal\_mul, [2319](#)
- nppsMul\_32sc\_Sfs
  - signal\_mul, [2319](#)
- nppsMul\_64f
  - signal\_mul, [2320](#)
- nppsMul\_64f\_I
  - signal\_mul, [2320](#)
- nppsMul\_64fc
  - signal\_mul, [2320](#)
- nppsMul\_64fc\_I
  - signal\_mul, [2321](#)
- nppsMul\_8u16u
  - signal\_mul, [2321](#)
- nppsMul\_8u\_ISfs
  - signal\_mul, [2321](#)
- nppsMul\_8u\_Sfs
  - signal\_mul, [2322](#)
- nppsMul\_Low\_32s\_Sfs
  - signal\_mul, [2322](#)
- nppsMulC\_16s\_ISfs
  - signal\_mulc, [2258](#)
- nppsMulC\_16s\_Sfs
  - signal\_mulc, [2259](#)
- nppsMulC\_16sc\_ISfs
  - signal\_mulc, [2259](#)
- nppsMulC\_16sc\_Sfs
  - signal\_mulc, [2259](#)
- nppsMulC\_16u\_ISfs
  - signal\_mulc, [2260](#)
- nppsMulC\_16u\_Sfs
  - signal\_mulc, [2260](#)
- nppsMulC\_32f
  - signal\_mulc, [2260](#)
- nppsMulC\_32f16s\_Sfs
  - signal\_mulc, [2261](#)
- nppsMulC\_32f\_I
  - signal\_mulc, [2261](#)
- nppsMulC\_32fc
  - signal\_mulc, [2261](#)
- nppsMulC\_32fc\_I
  - signal\_mulc, [2262](#)

- nppsMulC\_32s\_ISfs
  - signal\_mulc, [2262](#)
- nppsMulC\_32s\_Sfs
  - signal\_mulc, [2262](#)
- nppsMulC\_32sc\_ISfs
  - signal\_mulc, [2263](#)
- nppsMulC\_32sc\_Sfs
  - signal\_mulc, [2263](#)
- nppsMulC\_64f
  - signal\_mulc, [2263](#)
- nppsMulC\_64f64s\_ISfs
  - signal\_mulc, [2264](#)
- nppsMulC\_64f\_I
  - signal\_mulc, [2264](#)
- nppsMulC\_64fc
  - signal\_mulc, [2264](#)
- nppsMulC\_64fc\_I
  - signal\_mulc, [2265](#)
- nppsMulC\_8u\_ISfs
  - signal\_mulc, [2265](#)
- nppsMulC\_8u\_Sfs
  - signal\_mulc, [2265](#)
- nppsMulC\_Low\_32f16s
  - signal\_mulc, [2266](#)
- nppsNorm\_Inf\_16s32f
  - signal\_infinity\_norm, [2515](#)
- nppsNorm\_Inf\_16s32s\_Sfs
  - signal\_infinity\_norm, [2515](#)
- nppsNorm\_Inf\_32f
  - signal\_infinity\_norm, [2515](#)
- nppsNorm\_Inf\_32fc32f
  - signal\_infinity\_norm, [2515](#)
- nppsNorm\_Inf\_64f
  - signal\_infinity\_norm, [2516](#)
- nppsNorm\_Inf\_64fc64f
  - signal\_infinity\_norm, [2516](#)
- nppsNorm\_L1\_16s32f
  - signal\_L1\_norm, [2520](#)
- nppsNorm\_L1\_16s32s\_Sfs
  - signal\_L1\_norm, [2520](#)
- nppsNorm\_L1\_16s64s\_Sfs
  - signal\_L1\_norm, [2520](#)
- nppsNorm\_L1\_32f
  - signal\_L1\_norm, [2521](#)
- nppsNorm\_L1\_32fc64f
  - signal\_L1\_norm, [2521](#)
- nppsNorm\_L1\_64f
  - signal\_L1\_norm, [2521](#)
- nppsNorm\_L1\_64fc64f
  - signal\_L1\_norm, [2522](#)
- nppsNorm\_L2\_16s32f
  - signal\_L2\_norm, [2526](#)
- nppsNorm\_L2\_16s32s\_Sfs
  - signal\_L2\_norm, [2526](#)
- nppsNorm\_L2\_32f
  - signal\_L2\_norm, [2526](#)
- nppsNorm\_L2\_32fc64f
  - signal\_L2\_norm, [2527](#)
- nppsNorm\_L2\_64f
  - signal\_L2\_norm, [2527](#)
- nppsNorm\_L2\_64fc64f
  - signal\_L2\_norm, [2527](#)
- nppsNorm\_L2Sqr\_16s64s\_Sfs
  - signal\_L2\_norm, [2528](#)
- nppsNormalize\_16s\_Sfs
  - signal\_normalize, [2377](#)
- nppsNormalize\_16sc\_Sfs
  - signal\_normalize, [2378](#)
- nppsNormalize\_32f
  - signal\_normalize, [2378](#)
- nppsNormalize\_32fc
  - signal\_normalize, [2378](#)
- nppsNormalize\_64f
  - signal\_normalize, [2379](#)
- nppsNormalize\_64fc
  - signal\_normalize, [2379](#)
- nppsNormDiff\_Inf\_16s32f
  - signal\_infinity\_norm\_diff, [2532](#)
- nppsNormDiff\_Inf\_16s32s\_Sfs
  - signal\_infinity\_norm\_diff, [2532](#)
- nppsNormDiff\_Inf\_32f
  - signal\_infinity\_norm\_diff, [2532](#)
- nppsNormDiff\_Inf\_32fc32f
  - signal\_infinity\_norm\_diff, [2533](#)
- nppsNormDiff\_Inf\_64f
  - signal\_infinity\_norm\_diff, [2533](#)
- nppsNormDiff\_Inf\_64fc64f
  - signal\_infinity\_norm\_diff, [2533](#)
- nppsNormDiff\_L1\_16s32f
  - signal\_L1\_norm\_diff, [2537](#)
- nppsNormDiff\_L1\_16s32s\_Sfs
  - signal\_L1\_norm\_diff, [2537](#)
- nppsNormDiff\_L1\_16s64s\_Sfs
  - signal\_L1\_norm\_diff, [2537](#)
- nppsNormDiff\_L1\_32f
  - signal\_L1\_norm\_diff, [2538](#)
- nppsNormDiff\_L1\_32fc64f
  - signal\_L1\_norm\_diff, [2538](#)
- nppsNormDiff\_L1\_64f
  - signal\_L1\_norm\_diff, [2538](#)
- nppsNormDiff\_L1\_64fc64f
  - signal\_L1\_norm\_diff, [2539](#)
- nppsNormDiff\_L2\_16s32f
  - signal\_L2\_norm\_diff, [2543](#)
- nppsNormDiff\_L2\_16s32s\_Sfs
  - signal\_L2\_norm\_diff, [2543](#)
- nppsNormDiff\_L2\_32f
  - signal\_L2\_norm\_diff, [2543](#)



- nppsNormDiff\_L2\_32fc64f
  - signal\_L2\_norm\_diff, [2544](#)
- nppsNormDiff\_L2\_64f
  - signal\_L2\_norm\_diff, [2544](#)
- nppsNormDiff\_L2\_64fc64f
  - signal\_L2\_norm\_diff, [2544](#)
- nppsNormDiff\_L2Sqr\_16s64s\_Sfs
  - signal\_L2\_norm\_diff, [2545](#)
- nppsNormDiffInfGetBufferSize\_16s32f
  - signal\_infinity\_norm\_diff, [2534](#)
- nppsNormDiffInfGetBufferSize\_16s32s\_Sfs
  - signal\_infinity\_norm\_diff, [2534](#)
- nppsNormDiffInfGetBufferSize\_32f
  - signal\_infinity\_norm\_diff, [2534](#)
- nppsNormDiffInfGetBufferSize\_32fc32f
  - signal\_infinity\_norm\_diff, [2535](#)
- nppsNormDiffInfGetBufferSize\_64f
  - signal\_infinity\_norm\_diff, [2535](#)
- nppsNormDiffInfGetBufferSize\_64fc64f
  - signal\_infinity\_norm\_diff, [2535](#)
- nppsNormDiffL1GetBufferSize\_16s32f
  - signal\_L1\_norm\_diff, [2539](#)
- nppsNormDiffL1GetBufferSize\_16s32s\_Sfs
  - signal\_L1\_norm\_diff, [2539](#)
- nppsNormDiffL1GetBufferSize\_16s64s\_Sfs
  - signal\_L1\_norm\_diff, [2540](#)
- nppsNormDiffL1GetBufferSize\_32f
  - signal\_L1\_norm\_diff, [2540](#)
- nppsNormDiffL1GetBufferSize\_32fc64f
  - signal\_L1\_norm\_diff, [2540](#)
- nppsNormDiffL1GetBufferSize\_64f
  - signal\_L1\_norm\_diff, [2540](#)
- nppsNormDiffL1GetBufferSize\_64fc64f
  - signal\_L1\_norm\_diff, [2541](#)
- nppsNormDiffL2GetBufferSize\_16s32f
  - signal\_L2\_norm\_diff, [2545](#)
- nppsNormDiffL2GetBufferSize\_16s32s\_Sfs
  - signal\_L2\_norm\_diff, [2545](#)
- nppsNormDiffL2GetBufferSize\_32f
  - signal\_L2\_norm\_diff, [2546](#)
- nppsNormDiffL2GetBufferSize\_32fc64f
  - signal\_L2\_norm\_diff, [2546](#)
- nppsNormDiffL2GetBufferSize\_64f
  - signal\_L2\_norm\_diff, [2546](#)
- nppsNormDiffL2GetBufferSize\_64fc64f
  - signal\_L2\_norm\_diff, [2546](#)
- nppsNormDiffL2SqrGetBufferSize\_16s64s\_Sfs
  - signal\_L2\_norm\_diff, [2547](#)
- nppsNormInfGetBufferSize\_16s32f
  - signal\_infinity\_norm, [2516](#)
- nppsNormInfGetBufferSize\_16s32s\_Sfs
  - signal\_infinity\_norm, [2517](#)
- nppsNormInfGetBufferSize\_32f
  - signal\_infinity\_norm, [2517](#)
- nppsNormInfGetBufferSize\_32fc32f
  - signal\_infinity\_norm, [2517](#)
- nppsNormInfGetBufferSize\_64f
  - signal\_infinity\_norm, [2517](#)
- nppsNormInfGetBufferSize\_64fc64f
  - signal\_infinity\_norm, [2518](#)
- nppsNormL1GetBufferSize\_16s32f
  - signal\_L1\_norm, [2522](#)
- nppsNormL1GetBufferSize\_16s32s\_Sfs
  - signal\_L1\_norm, [2522](#)
- nppsNormL1GetBufferSize\_16s64s\_Sfs
  - signal\_L1\_norm, [2522](#)
- nppsNormL1GetBufferSize\_32f
  - signal\_L1\_norm, [2523](#)
- nppsNormL1GetBufferSize\_32fc64f
  - signal\_L1\_norm, [2523](#)
- nppsNormL1GetBufferSize\_64f
  - signal\_L1\_norm, [2523](#)
- nppsNormL1GetBufferSize\_64fc64f
  - signal\_L1\_norm, [2523](#)
- nppsNormL2GetBufferSize\_16s32f
  - signal\_L2\_norm, [2528](#)
- nppsNormL2GetBufferSize\_16s32s\_Sfs
  - signal\_L2\_norm, [2528](#)
- nppsNormL2GetBufferSize\_32f
  - signal\_L2\_norm, [2528](#)
- nppsNormL2GetBufferSize\_32fc64f
  - signal\_L2\_norm, [2529](#)
- nppsNormL2GetBufferSize\_64f
  - signal\_L2\_norm, [2529](#)
- nppsNormL2GetBufferSize\_64fc64f
  - signal\_L2\_norm, [2529](#)
- nppsNormL2SqrGetBufferSize\_16s64s\_Sfs
  - signal\_L2\_norm, [2529](#)
- nppsNot\_16u
  - signal\_not, [2401](#)
- nppsNot\_16u\_I
  - signal\_not, [2401](#)
- nppsNot\_32u
  - signal\_not, [2402](#)
- nppsNot\_32u\_I
  - signal\_not, [2402](#)
- nppsNot\_8u
  - signal\_not, [2402](#)
- nppsNot\_8u\_I
  - signal\_not, [2402](#)
- nppsOr\_16u
  - signal\_or, [2392](#)
- nppsOr\_16u\_I
  - signal\_or, [2392](#)
- nppsOr\_32u
  - signal\_or, [2393](#)
- nppsOr\_32u\_I
  - signal\_or, [2393](#)

- nppsOr\_8u
  - signal\_or, [2393](#)
- nppsOr\_8u\_I
  - signal\_or, [2394](#)
- nppsOrC\_16u
  - signal\_orc, [2389](#)
- nppsOrC\_16u\_I
  - signal\_orc, [2389](#)
- nppsOrC\_32u
  - signal\_orc, [2390](#)
- nppsOrC\_32u\_I
  - signal\_orc, [2390](#)
- nppsOrC\_8u
  - signal\_orc, [2390](#)
- nppsOrC\_8u\_I
  - signal\_orc, [2391](#)
- nppsRShiftC\_16s
  - signal\_rshifc, [2408](#)
- nppsRShiftC\_16s\_I
  - signal\_rshifc, [2409](#)
- nppsRShiftC\_16u
  - signal\_rshifc, [2409](#)
- nppsRShiftC\_16u\_I
  - signal\_rshifc, [2409](#)
- nppsRShiftC\_32s
  - signal\_rshifc, [2409](#)
- nppsRShiftC\_32s\_I
  - signal\_rshifc, [2410](#)
- nppsRShiftC\_32u
  - signal\_rshifc, [2410](#)
- nppsRShiftC\_32u\_I
  - signal\_rshifc, [2410](#)
- nppsRShiftC\_8u
  - signal\_rshifc, [2411](#)
- nppsRShiftC\_8u\_I
  - signal\_rshifc, [2411](#)
- nppsSet\_16s
  - signal\_set, [2445](#)
- nppsSet\_16sc
  - signal\_set, [2445](#)
- nppsSet\_16u
  - signal\_set, [2445](#)
- nppsSet\_32f
  - signal\_set, [2445](#)
- nppsSet\_32fc
  - signal\_set, [2446](#)
- nppsSet\_32s
  - signal\_set, [2446](#)
- nppsSet\_32sc
  - signal\_set, [2446](#)
- nppsSet\_32u
  - signal\_set, [2446](#)
- nppsSet\_64f
  - signal\_set, [2447](#)
- nppsSet\_64fc
  - signal\_set, [2447](#)
- nppsSet\_64s
  - signal\_set, [2447](#)
- nppsSet\_64sc
  - signal\_set, [2448](#)
- nppsSet\_8s
  - signal\_set, [2448](#)
- nppsSet\_8u
  - signal\_set, [2448](#)
- nppsSqr\_16s\_ISfs
  - signal\_square, [2348](#)
- nppsSqr\_16s\_Sfs
  - signal\_square, [2348](#)
- nppsSqr\_16sc\_ISfs
  - signal\_square, [2348](#)
- nppsSqr\_16sc\_Sfs
  - signal\_square, [2349](#)
- nppsSqr\_16u\_ISfs
  - signal\_square, [2349](#)
- nppsSqr\_16u\_Sfs
  - signal\_square, [2349](#)
- nppsSqr\_32f
  - signal\_square, [2349](#)
- nppsSqr\_32f\_I
  - signal\_square, [2350](#)
- nppsSqr\_32fc
  - signal\_square, [2350](#)
- nppsSqr\_32fc\_I
  - signal\_square, [2350](#)
- nppsSqr\_64f
  - signal\_square, [2350](#)
- nppsSqr\_64f\_I
  - signal\_square, [2351](#)
- nppsSqr\_64fc
  - signal\_square, [2351](#)
- nppsSqr\_64fc\_I
  - signal\_square, [2351](#)
- nppsSqr\_8u\_ISfs
  - signal\_square, [2351](#)
- nppsSqr\_8u\_Sfs
  - signal\_square, [2352](#)
- nppsSqrt\_16s\_ISfs
  - signal\_sqrt, [2354](#)
- nppsSqrt\_16s\_Sfs
  - signal\_sqrt, [2354](#)
- nppsSqrt\_16sc\_ISfs
  - signal\_sqrt, [2355](#)
- nppsSqrt\_16sc\_Sfs
  - signal\_sqrt, [2355](#)
- nppsSqrt\_16u\_ISfs
  - signal\_sqrt, [2355](#)
- nppsSqrt\_16u\_Sfs
  - signal\_sqrt, [2355](#)

- nppsSqrt\_32f
  - signal\_sqrt, [2356](#)
- nppsSqrt\_32f\_I
  - signal\_sqrt, [2356](#)
- nppsSqrt\_32fc
  - signal\_sqrt, [2356](#)
- nppsSqrt\_32fc\_I
  - signal\_sqrt, [2357](#)
- nppsSqrt\_32s16s\_Sfs
  - signal\_sqrt, [2357](#)
- nppsSqrt\_64f
  - signal\_sqrt, [2357](#)
- nppsSqrt\_64f\_I
  - signal\_sqrt, [2357](#)
- nppsSqrt\_64fc
  - signal\_sqrt, [2358](#)
- nppsSqrt\_64fc\_I
  - signal\_sqrt, [2358](#)
- nppsSqrt\_64s16s\_Sfs
  - signal\_sqrt, [2358](#)
- nppsSqrt\_64s\_ISfs
  - signal\_sqrt, [2358](#)
- nppsSqrt\_64s\_Sfs
  - signal\_sqrt, [2359](#)
- nppsSqrt\_8u\_ISfs
  - signal\_sqrt, [2359](#)
- nppsSqrt\_8u\_Sfs
  - signal\_sqrt, [2359](#)
- nppsStdDev\_16s32s\_Sfs
  - signal\_standard\_deviation, [2495](#)
- nppsStdDev\_16s\_Sfs
  - signal\_standard\_deviation, [2495](#)
- nppsStdDev\_32f
  - signal\_standard\_deviation, [2496](#)
- nppsStdDev\_64f
  - signal\_standard\_deviation, [2496](#)
- nppsStdDevGetBufferSize\_16s32s\_Sfs
  - signal\_standard\_deviation, [2496](#)
- nppsStdDevGetBufferSize\_16s\_Sfs
  - signal\_standard\_deviation, [2497](#)
- nppsStdDevGetBufferSize\_32f
  - signal\_standard\_deviation, [2497](#)
- nppsStdDevGetBufferSize\_64f
  - signal\_standard\_deviation, [2497](#)
- nppsSub\_16s
  - signal\_sub, [2324](#)
- nppsSub\_16s32f
  - signal\_sub, [2325](#)
- nppsSub\_16s\_I
  - signal\_sub, [2325](#)
- nppsSub\_16s\_ISfs
  - signal\_sub, [2325](#)
- nppsSub\_16s\_Sfs
  - signal\_sub, [2326](#)
- nppsSub\_16sc\_ISfs
  - signal\_sub, [2326](#)
- nppsSub\_16sc\_Sfs
  - signal\_sub, [2326](#)
- nppsSub\_16u\_ISfs
  - signal\_sub, [2327](#)
- nppsSub\_16u\_Sfs
  - signal\_sub, [2327](#)
- nppsSub\_32f
  - signal\_sub, [2327](#)
- nppsSub\_32f\_I
  - signal\_sub, [2328](#)
- nppsSub\_32fc
  - signal\_sub, [2328](#)
- nppsSub\_32fc\_I
  - signal\_sub, [2328](#)
- nppsSub\_32s\_ISfs
  - signal\_sub, [2328](#)
- nppsSub\_32s\_Sfs
  - signal\_sub, [2329](#)
- nppsSub\_32sc\_ISfs
  - signal\_sub, [2329](#)
- nppsSub\_32sc\_Sfs
  - signal\_sub, [2329](#)
- nppsSub\_64f
  - signal\_sub, [2330](#)
- nppsSub\_64f\_I
  - signal\_sub, [2330](#)
- nppsSub\_64fc
  - signal\_sub, [2330](#)
- nppsSub\_64fc\_I
  - signal\_sub, [2331](#)
- nppsSub\_8u\_ISfs
  - signal\_sub, [2331](#)
- nppsSub\_8u\_Sfs
  - signal\_sub, [2331](#)
- nppsSubC\_16s\_ISfs
  - signal\_subc, [2268](#)
- nppsSubC\_16s\_Sfs
  - signal\_subc, [2268](#)
- nppsSubC\_16sc\_ISfs
  - signal\_subc, [2269](#)
- nppsSubC\_16sc\_Sfs
  - signal\_subc, [2269](#)
- nppsSubC\_16u\_ISfs
  - signal\_subc, [2269](#)
- nppsSubC\_16u\_Sfs
  - signal\_subc, [2270](#)
- nppsSubC\_32f
  - signal\_subc, [2270](#)
- nppsSubC\_32f\_I
  - signal\_subc, [2270](#)
- nppsSubC\_32fc
  - signal\_subc, [2271](#)

- signal\_subc, [2271](#)
- nppsSubC\_32s\_ISfs
  - signal\_subc, [2271](#)
- nppsSubC\_32s\_Sfs
  - signal\_subc, [2272](#)
- nppsSubC\_32sc\_ISfs
  - signal\_subc, [2272](#)
- nppsSubC\_32sc\_Sfs
  - signal\_subc, [2272](#)
- nppsSubC\_64f
  - signal\_subc, [2273](#)
- nppsSubC\_64f\_I
  - signal\_subc, [2273](#)
- nppsSubC\_64fc
  - signal\_subc, [2273](#)
- nppsSubC\_64fc\_I
  - signal\_subc, [2274](#)
- nppsSubC\_8u\_ISfs
  - signal\_subc, [2274](#)
- nppsSubC\_8u\_Sfs
  - signal\_subc, [2274](#)
- nppsSubCRev\_16s\_ISfs
  - signal\_subcrev, [2277](#)
- nppsSubCRev\_16s\_Sfs
  - signal\_subcrev, [2278](#)
- nppsSubCRev\_16sc\_ISfs
  - signal\_subcrev, [2278](#)
- nppsSubCRev\_16sc\_Sfs
  - signal\_subcrev, [2278](#)
- nppsSubCRev\_16u\_ISfs
  - signal\_subcrev, [2279](#)
- nppsSubCRev\_16u\_Sfs
  - signal\_subcrev, [2279](#)
- nppsSubCRev\_32f
  - signal\_subcrev, [2279](#)
- nppsSubCRev\_32f\_I
  - signal\_subcrev, [2280](#)
- nppsSubCRev\_32fc
  - signal\_subcrev, [2280](#)
- nppsSubCRev\_32fc\_I
  - signal\_subcrev, [2280](#)
- nppsSubCRev\_32s\_ISfs
  - signal\_subcrev, [2280](#)
- nppsSubCRev\_32s\_Sfs
  - signal\_subcrev, [2281](#)
- nppsSubCRev\_32sc\_ISfs
  - signal\_subcrev, [2281](#)
- nppsSubCRev\_32sc\_Sfs
  - signal\_subcrev, [2281](#)
- nppsSubCRev\_64f
  - signal\_subcrev, [2282](#)
- nppsSubCRev\_64f\_I
  - signal\_subcrev, [2282](#)
- nppsSubCRev\_64fc
  - signal\_subcrev, [2282](#)
- nppsSubCRev\_64fc\_I
  - signal\_subcrev, [2283](#)
- nppsSubCRev\_8u\_ISfs
  - signal\_subcrev, [2283](#)
- nppsSubCRev\_8u\_Sfs
  - signal\_subcrev, [2283](#)
- nppsSum\_16s32s\_Sfs
  - signal\_sum, [2463](#)
- nppsSum\_16s\_Sfs
  - signal\_sum, [2463](#)
- nppsSum\_16sc32sc\_Sfs
  - signal\_sum, [2464](#)
- nppsSum\_16sc\_Sfs
  - signal\_sum, [2464](#)
- nppsSum\_32f
  - signal\_sum, [2464](#)
- nppsSum\_32fc
  - signal\_sum, [2465](#)
- nppsSum\_32s\_Sfs
  - signal\_sum, [2465](#)
- nppsSum\_64f
  - signal\_sum, [2465](#)
- nppsSum\_64fc
  - signal\_sum, [2466](#)
- nppsSumGetBufferSize\_16s32s\_Sfs
  - signal\_sum, [2466](#)
- nppsSumGetBufferSize\_16s\_Sfs
  - signal\_sum, [2466](#)
- nppsSumGetBufferSize\_16sc32sc\_Sfs
  - signal\_sum, [2467](#)
- nppsSumGetBufferSize\_16sc\_Sfs
  - signal\_sum, [2467](#)
- nppsSumGetBufferSize\_32f
  - signal\_sum, [2467](#)
- nppsSumGetBufferSize\_32fc
  - signal\_sum, [2467](#)
- nppsSumGetBufferSize\_32s\_Sfs
  - signal\_sum, [2468](#)
- nppsSumGetBufferSize\_64f
  - signal\_sum, [2468](#)
- nppsSumGetBufferSize\_64fc
  - signal\_sum, [2468](#)
- nppsSumLn\_16s32f
  - signal\_sumln, [2371](#)
- nppsSumLn\_32f
  - signal\_sumln, [2372](#)
- nppsSumLn\_32f64f
  - signal\_sumln, [2372](#)
- nppsSumLn\_64f
  - signal\_sumln, [2372](#)
- nppsSumLnGetBufferSize\_16s32f
  - signal\_sumln, [2373](#)

- nppsSumLnGetBufferSize\_32f
  - signal\_sumln, [2373](#)
- nppsSumLnGetBufferSize\_32f64f
  - signal\_sumln, [2373](#)
- nppsSumLnGetBufferSize\_64f
  - signal\_sumln, [2373](#)
- NppStatus
  - typedefs\_npp, [44](#)
- nppsThreshold\_16s
  - signal\_threshold, [2420](#)
- nppsThreshold\_16s\_I
  - signal\_threshold, [2421](#)
- nppsThreshold\_16sc
  - signal\_threshold, [2421](#)
- nppsThreshold\_16sc\_I
  - signal\_threshold, [2421](#)
- nppsThreshold\_32f
  - signal\_threshold, [2422](#)
- nppsThreshold\_32f\_I
  - signal\_threshold, [2422](#)
- nppsThreshold\_32fc
  - signal\_threshold, [2422](#)
- nppsThreshold\_32fc\_I
  - signal\_threshold, [2423](#)
- nppsThreshold\_64f
  - signal\_threshold, [2423](#)
- nppsThreshold\_64f\_I
  - signal\_threshold, [2423](#)
- nppsThreshold\_64fc
  - signal\_threshold, [2424](#)
- nppsThreshold\_64fc\_I
  - signal\_threshold, [2424](#)
- nppsThreshold\_GT\_16s
  - signal\_threshold, [2424](#)
- nppsThreshold\_GT\_16s\_I
  - signal\_threshold, [2425](#)
- nppsThreshold\_GT\_16sc
  - signal\_threshold, [2425](#)
- nppsThreshold\_GT\_16sc\_I
  - signal\_threshold, [2425](#)
- nppsThreshold\_GT\_32f
  - signal\_threshold, [2426](#)
- nppsThreshold\_GT\_32f\_I
  - signal\_threshold, [2426](#)
- nppsThreshold\_GT\_32fc
  - signal\_threshold, [2426](#)
- nppsThreshold\_GT\_32fc\_I
  - signal\_threshold, [2427](#)
- nppsThreshold\_GT\_64f
  - signal\_threshold, [2427](#)
- nppsThreshold\_GT\_64f\_I
  - signal\_threshold, [2427](#)
- nppsThreshold\_GT\_64fc
  - signal\_threshold, [2428](#)
- nppsThreshold\_GT\_64fc\_I
  - signal\_threshold, [2428](#)
- nppsThreshold\_GTVal\_16s
  - signal\_threshold, [2428](#)
- nppsThreshold\_GTVal\_16s\_I
  - signal\_threshold, [2429](#)
- nppsThreshold\_GTVal\_16sc
  - signal\_threshold, [2429](#)
- nppsThreshold\_GTVal\_16sc\_I
  - signal\_threshold, [2429](#)
- nppsThreshold\_GTVal\_32f
  - signal\_threshold, [2430](#)
- nppsThreshold\_GTVal\_32f\_I
  - signal\_threshold, [2430](#)
- nppsThreshold\_GTVal\_32fc
  - signal\_threshold, [2430](#)
- nppsThreshold\_GTVal\_32fc\_I
  - signal\_threshold, [2431](#)
- nppsThreshold\_GTVal\_64f
  - signal\_threshold, [2431](#)
- nppsThreshold\_GTVal\_64f\_I
  - signal\_threshold, [2431](#)
- nppsThreshold\_GTVal\_64fc
  - signal\_threshold, [2432](#)
- nppsThreshold\_GTVal\_64fc\_I
  - signal\_threshold, [2432](#)
- nppsThreshold\_LT\_16s
  - signal\_threshold, [2432](#)
- nppsThreshold\_LT\_16s\_I
  - signal\_threshold, [2433](#)
- nppsThreshold\_LT\_16sc
  - signal\_threshold, [2433](#)
- nppsThreshold\_LT\_16sc\_I
  - signal\_threshold, [2433](#)
- nppsThreshold\_LT\_32f
  - signal\_threshold, [2434](#)
- nppsThreshold\_LT\_32f\_I
  - signal\_threshold, [2434](#)
- nppsThreshold\_LT\_32fc
  - signal\_threshold, [2434](#)
- nppsThreshold\_LT\_32fc\_I
  - signal\_threshold, [2435](#)
- nppsThreshold\_LT\_64f
  - signal\_threshold, [2435](#)
- nppsThreshold\_LT\_64f\_I
  - signal\_threshold, [2435](#)
- nppsThreshold\_LT\_64fc
  - signal\_threshold, [2436](#)
- nppsThreshold\_LT\_64fc\_I
  - signal\_threshold, [2436](#)
- nppsThreshold\_LTVa\_16s
  - signal\_threshold, [2436](#)
- nppsThreshold\_LTVa\_16s\_I
  - signal\_threshold, [2437](#)

- nppsThreshold\_LTVa1\_16sc
  - signal\_threshold, [2437](#)
- nppsThreshold\_LTVa1\_16sc\_I
  - signal\_threshold, [2437](#)
- nppsThreshold\_LTVa1\_32f
  - signal\_threshold, [2438](#)
- nppsThreshold\_LTVa1\_32f\_I
  - signal\_threshold, [2438](#)
- nppsThreshold\_LTVa1\_32fc
  - signal\_threshold, [2438](#)
- nppsThreshold\_LTVa1\_32fc\_I
  - signal\_threshold, [2439](#)
- nppsThreshold\_LTVa1\_64f
  - signal\_threshold, [2439](#)
- nppsThreshold\_LTVa1\_64f\_I
  - signal\_threshold, [2439](#)
- nppsThreshold\_LTVa1\_64fc
  - signal\_threshold, [2440](#)
- nppsThreshold\_LTVa1\_64fc\_I
  - signal\_threshold, [2440](#)
- nppsXor\_16u
  - signal\_xor, [2398](#)
- nppsXor\_16u\_I
  - signal\_xor, [2398](#)
- nppsXor\_32u
  - signal\_xor, [2399](#)
- nppsXor\_32u\_I
  - signal\_xor, [2399](#)
- nppsXor\_8u
  - signal\_xor, [2399](#)
- nppsXor\_8u\_I
  - signal\_xor, [2400](#)
- nppsXorC\_16u
  - signal\_xorc, [2395](#)
- nppsXorC\_16u\_I
  - signal\_xorc, [2395](#)
- nppsXorC\_32u
  - signal\_xorc, [2396](#)
- nppsXorC\_32u\_I
  - signal\_xorc, [2396](#)
- nppsXorC\_8u
  - signal\_xorc, [2396](#)
- nppsXorC\_8u\_I
  - signal\_xorc, [2397](#)
- NppsZCType
  - typedefs\_npp, [46](#)
- nppsZero\_16s
  - signal\_zero, [2449](#)
- nppsZero\_16sc
  - signal\_zero, [2450](#)
- nppsZero\_32f
  - signal\_zero, [2450](#)
- nppsZero\_32fc
  - signal\_zero, [2450](#)
- nppsZero\_32s
  - signal\_zero, [2450](#)
- nppsZero\_32sc
  - signal\_zero, [2450](#)
- nppsZero\_64f
  - signal\_zero, [2451](#)
- nppsZero\_64fc
  - signal\_zero, [2451](#)
- nppsZero\_64s
  - signal\_zero, [2451](#)
- nppsZero\_64sc
  - signal\_zero, [2451](#)
- nppsZero\_8u
  - signal\_zero, [2452](#)
- nppsZeroCrossing\_16s32f
  - signal\_count\_zero\_crossings, [2569](#)
- nppsZeroCrossing\_32f
  - signal\_count\_zero\_crossings, [2569](#)
- nppsZeroCrossingGetBufferSize\_16s32f
  - signal\_count\_zero\_crossings, [2570](#)
- nppsZeroCrossingGetBufferSize\_32f
  - signal\_count\_zero\_crossings, [2570](#)
- nppZCC
  - typedefs\_npp, [46](#)
- nppZCR
  - typedefs\_npp, [46](#)
- nppZCXor
  - typedefs\_npp, [46](#)
- numClassifiers
  - NppiHaarClassifier\_32f, [2630](#)
- Or, [444](#), [2392](#)
- OrC, [382](#), [2389](#)
- Perspective Transform, [1317](#)
- Quantization Functions, [705](#)
- Rank Filters, [1121](#)
- re
  - NPP\_ALIGN\_16, [2626](#)
  - NPP\_ALIGN\_8, [2627](#), [2628](#)
- RectStdDev, [1851](#)
- Remap, [1219](#)
- Resize, [1207](#)
- ResizeSqrPixel, [1185](#)
- Rotate, [1241](#)
- RShiftC, [404](#), [2408](#)
- Scale, [845](#)
- Set, [720](#), [2444](#)
- signal\_10log10
  - npps10Log10\_32s\_ISfs, [2370](#)
  - npps10Log10\_32s\_Sfs, [2370](#)
- signal\_abs

- nppsAbs\_16s, [2344](#)
- nppsAbs\_16s\_I, [2344](#)
- nppsAbs\_32f, [2345](#)
- nppsAbs\_32f\_I, [2345](#)
- nppsAbs\_32s, [2345](#)
- nppsAbs\_32s\_I, [2345](#)
- nppsAbs\_64f, [2346](#)
- nppsAbs\_64f\_I, [2346](#)
- signal\_add
  - nppsAdd\_16s, [2296](#)
  - nppsAdd\_16s32f, [2296](#)
  - nppsAdd\_16s32s\_I, [2296](#)
  - nppsAdd\_16s\_I, [2297](#)
  - nppsAdd\_16s\_ISfs, [2297](#)
  - nppsAdd\_16s\_Sfs, [2297](#)
  - nppsAdd\_16sc\_ISfs, [2298](#)
  - nppsAdd\_16sc\_Sfs, [2298](#)
  - nppsAdd\_16u, [2298](#)
  - nppsAdd\_16u\_ISfs, [2299](#)
  - nppsAdd\_16u\_Sfs, [2299](#)
  - nppsAdd\_32f, [2299](#)
  - nppsAdd\_32f\_I, [2300](#)
  - nppsAdd\_32fc, [2300](#)
  - nppsAdd\_32fc\_I, [2300](#)
  - nppsAdd\_32s\_ISfs, [2301](#)
  - nppsAdd\_32s\_Sfs, [2301](#)
  - nppsAdd\_32sc\_ISfs, [2301](#)
  - nppsAdd\_32sc\_Sfs, [2302](#)
  - nppsAdd\_32u, [2302](#)
  - nppsAdd\_64f, [2302](#)
  - nppsAdd\_64f\_I, [2303](#)
  - nppsAdd\_64fc, [2303](#)
  - nppsAdd\_64fc\_I, [2303](#)
  - nppsAdd\_64s\_Sfs, [2304](#)
  - nppsAdd\_8u16u, [2304](#)
  - nppsAdd\_8u\_ISfs, [2304](#)
  - nppsAdd\_8u\_Sfs, [2305](#)
- signal\_addc
  - nppsAddC\_16s\_ISfs, [2248](#)
  - nppsAddC\_16s\_Sfs, [2248](#)
  - nppsAddC\_16sc\_ISfs, [2249](#)
  - nppsAddC\_16sc\_Sfs, [2249](#)
  - nppsAddC\_16u\_ISfs, [2249](#)
  - nppsAddC\_16u\_Sfs, [2250](#)
  - nppsAddC\_32f, [2250](#)
  - nppsAddC\_32f\_I, [2250](#)
  - nppsAddC\_32fc, [2251](#)
  - nppsAddC\_32fc\_I, [2251](#)
  - nppsAddC\_32s\_ISfs, [2251](#)
  - nppsAddC\_32s\_Sfs, [2252](#)
  - nppsAddC\_32sc\_ISfs, [2252](#)
  - nppsAddC\_32sc\_Sfs, [2252](#)
  - nppsAddC\_64f, [2253](#)
  - nppsAddC\_64f\_I, [2253](#)
  - nppsAddC\_64fc, [2253](#)
  - nppsAddC\_64fc\_I, [2254](#)
  - nppsAddC\_8u\_ISfs, [2254](#)
  - nppsAddC\_8u\_Sfs, [2254](#)
- signal\_addproduct
  - nppsAddProduct\_16s32s\_Sfs, [2307](#)
  - nppsAddProduct\_16s\_Sfs, [2307](#)
  - nppsAddProduct\_32f, [2307](#)
  - nppsAddProduct\_32fc, [2308](#)
  - nppsAddProduct\_32s\_Sfs, [2308](#)
  - nppsAddProduct\_64f, [2308](#)
  - nppsAddProduct\_64fc, [2309](#)
- signal\_addproductc
  - nppsAddProductC\_32f, [2256](#)
- signal\_and
  - nppsAnd\_16u, [2386](#)
  - nppsAnd\_16u\_I, [2386](#)
  - nppsAnd\_32u, [2387](#)
  - nppsAnd\_32u\_I, [2387](#)
  - nppsAnd\_8u, [2387](#)
  - nppsAnd\_8u\_I, [2388](#)
- signal\_andc
  - nppsAndC\_16u, [2383](#)
  - nppsAndC\_16u\_I, [2383](#)
  - nppsAndC\_32u, [2384](#)
  - nppsAndC\_32u\_I, [2384](#)
  - nppsAndC\_8u, [2384](#)
  - nppsAndC\_8u\_I, [2385](#)
- signal\_average\_error
  - nppsAverageError\_16s, [2584](#)
  - nppsAverageError\_16sc, [2584](#)
  - nppsAverageError\_16u, [2584](#)
  - nppsAverageError\_32f, [2585](#)
  - nppsAverageError\_32fc, [2585](#)
  - nppsAverageError\_32s, [2585](#)
  - nppsAverageError\_32sc, [2586](#)
  - nppsAverageError\_32u, [2586](#)
  - nppsAverageError\_64f, [2586](#)
  - nppsAverageError\_64fc, [2587](#)
  - nppsAverageError\_64s, [2587](#)
  - nppsAverageError\_64sc, [2587](#)
  - nppsAverageError\_8s, [2588](#)
  - nppsAverageError\_8u, [2588](#)
  - nppsAverageErrorGetBufferSize\_16s, [2588](#)
  - nppsAverageErrorGetBufferSize\_16sc, [2589](#)
  - nppsAverageErrorGetBufferSize\_16u, [2589](#)
  - nppsAverageErrorGetBufferSize\_32f, [2589](#)
  - nppsAverageErrorGetBufferSize\_32fc, [2589](#)
  - nppsAverageErrorGetBufferSize\_32s, [2590](#)
  - nppsAverageErrorGetBufferSize\_32sc, [2590](#)
  - nppsAverageErrorGetBufferSize\_32u, [2590](#)
  - nppsAverageErrorGetBufferSize\_64f, [2590](#)
  - nppsAverageErrorGetBufferSize\_64fc, [2591](#)
  - nppsAverageErrorGetBufferSize\_64s, [2591](#)



- nppsAverageErrorGetBufferSize\_64sc, 2591
- nppsAverageErrorGetBufferSize\_8s, 2591
- nppsAverageErrorGetBufferSize\_8u, 2592
- signal\_average\_relative\_error
  - nppsAverageRelativeError\_16s, 2607
  - nppsAverageRelativeError\_16sc, 2607
  - nppsAverageRelativeError\_16u, 2608
  - nppsAverageRelativeError\_32f, 2608
  - nppsAverageRelativeError\_32fc, 2608
  - nppsAverageRelativeError\_32s, 2609
  - nppsAverageRelativeError\_32sc, 2609
  - nppsAverageRelativeError\_32u, 2610
  - nppsAverageRelativeError\_64f, 2610
  - nppsAverageRelativeError\_64fc, 2610
  - nppsAverageRelativeError\_64s, 2611
  - nppsAverageRelativeError\_64sc, 2611
  - nppsAverageRelativeError\_8s, 2612
  - nppsAverageRelativeError\_8u, 2612
  - nppsAverageRelativeErrorGetBufferSize\_16s, 2612
  - nppsAverageRelativeErrorGetBufferSize\_-16sc, 2613
  - nppsAverageRelativeErrorGetBufferSize\_16u, 2613
  - nppsAverageRelativeErrorGetBufferSize\_32f, 2613
  - nppsAverageRelativeErrorGetBufferSize\_-32fc, 2613
  - nppsAverageRelativeErrorGetBufferSize\_32s, 2614
  - nppsAverageRelativeErrorGetBufferSize\_-32sc, 2614
  - nppsAverageRelativeErrorGetBufferSize\_32u, 2614
  - nppsAverageRelativeErrorGetBufferSize\_64f, 2614
  - nppsAverageRelativeErrorGetBufferSize\_-64fc, 2615
  - nppsAverageRelativeErrorGetBufferSize\_64s, 2615
  - nppsAverageRelativeErrorGetBufferSize\_-64sc, 2615
  - nppsAverageRelativeErrorGetBufferSize\_8s, 2615
  - nppsAverageRelativeErrorGetBufferSize\_8u, 2616
- signal\_cauchy
  - nppsCauchy\_32f\_I, 2380
  - nppsCauchyD\_32f\_I, 2380
  - nppsCauchyDD2\_32f\_I, 2380
- signal\_convert
  - nppsConvert\_16s32f, 2415
  - nppsConvert\_16s32f\_Sfs, 2415
  - nppsConvert\_16s32s, 2415
  - nppsConvert\_16s64f\_Sfs, 2415
  - nppsConvert\_16s8s\_Sfs, 2415
  - nppsConvert\_16u32f, 2415
  - nppsConvert\_32f16s\_Sfs, 2415
  - nppsConvert\_32f16u\_Sfs, 2415
  - nppsConvert\_32f32s\_Sfs, 2415
  - nppsConvert\_32f64f, 2415
  - nppsConvert\_32f8s\_Sfs, 2415
  - nppsConvert\_32f8u\_Sfs, 2415
  - nppsConvert\_32s16s, 2415
  - nppsConvert\_32s16s\_Sfs, 2415
  - nppsConvert\_32s32f, 2415
  - nppsConvert\_32s32f\_Sfs, 2415
  - nppsConvert\_32s64f, 2415
  - nppsConvert\_32s64f\_Sfs, 2415
  - nppsConvert\_64f16s\_Sfs, 2415
  - nppsConvert\_64f32f, 2415
  - nppsConvert\_64f32s\_Sfs, 2415
  - nppsConvert\_64f64s\_Sfs, 2415
  - nppsConvert\_64s32s\_Sfs, 2415
  - nppsConvert\_64s64f, 2415
  - nppsConvert\_8s16s, 2415
  - nppsConvert\_8s32f, 2415
  - nppsConvert\_8u32f, 2415
- signal\_copy
  - nppsCopy\_16s, 2453
  - nppsCopy\_16sc, 2454
  - nppsCopy\_32f, 2454
  - nppsCopy\_32fc, 2454
  - nppsCopy\_32s, 2454
  - nppsCopy\_32sc, 2455
  - nppsCopy\_64fc, 2455
  - nppsCopy\_64s, 2455
  - nppsCopy\_64sc, 2456
  - nppsCopy\_8u, 2456
- signal\_count\_in\_range
  - nppsCountInRange\_32s, 2568
  - nppsCountInRangeGetBufferSize\_32s, 2568
- signal\_count\_zero\_crossings
  - nppsZeroCrossing\_16s32f, 2569
  - nppsZeroCrossing\_32f, 2569
  - nppsZeroCrossingGetBufferSize\_16s32f, 2570
  - nppsZeroCrossingGetBufferSize\_32f, 2570
- signal\_cuberoot
  - nppsCubrt\_32f, 2361
  - nppsCubrt\_32s16s\_Sfs, 2361
- signal\_div
  - nppsDiv\_16s\_ISfs, 2334
  - nppsDiv\_16s\_Sfs, 2334
  - nppsDiv\_16sc\_ISfs, 2335
  - nppsDiv\_16sc\_Sfs, 2335
  - nppsDiv\_16u\_ISfs, 2335
  - nppsDiv\_16u\_Sfs, 2336



- nppsDiv\_32f, [2336](#)
- nppsDiv\_32f\_I, [2336](#)
- nppsDiv\_32fc, [2337](#)
- nppsDiv\_32fc\_I, [2337](#)
- nppsDiv\_32s16s\_Sfs, [2337](#)
- nppsDiv\_32s\_ISfs, [2338](#)
- nppsDiv\_32s\_Sfs, [2338](#)
- nppsDiv\_64f, [2338](#)
- nppsDiv\_64f\_I, [2339](#)
- nppsDiv\_64fc, [2339](#)
- nppsDiv\_64fc\_I, [2339](#)
- nppsDiv\_8u\_ISfs, [2340](#)
- nppsDiv\_8u\_Sfs, [2340](#)
- signal\_divc
  - nppsDivC\_16s\_ISfs, [2286](#)
  - nppsDivC\_16s\_Sfs, [2286](#)
  - nppsDivC\_16sc\_ISfs, [2286](#)
  - nppsDivC\_16sc\_Sfs, [2287](#)
  - nppsDivC\_16u\_ISfs, [2287](#)
  - nppsDivC\_16u\_Sfs, [2287](#)
  - nppsDivC\_32f, [2288](#)
  - nppsDivC\_32f\_I, [2288](#)
  - nppsDivC\_32fc, [2288](#)
  - nppsDivC\_32fc\_I, [2289](#)
  - nppsDivC\_64f, [2289](#)
  - nppsDivC\_64f\_I, [2289](#)
  - nppsDivC\_64fc, [2290](#)
  - nppsDivC\_64fc\_I, [2290](#)
  - nppsDivC\_8u\_ISfs, [2290](#)
  - nppsDivC\_8u\_Sfs, [2291](#)
- signal\_divcrev
  - nppsDivCRev\_16u, [2292](#)
  - nppsDivCRev\_16u\_I, [2292](#)
  - nppsDivCRev\_32f, [2293](#)
  - nppsDivCRev\_32f\_I, [2293](#)
- signal\_divround
  - nppsDiv\_Round\_16s\_ISfs, [2341](#)
  - nppsDiv\_Round\_16s\_Sfs, [2342](#)
  - nppsDiv\_Round\_16u\_ISfs, [2342](#)
  - nppsDiv\_Round\_16u\_Sfs, [2342](#)
  - nppsDiv\_Round\_8u\_ISfs, [2343](#)
  - nppsDiv\_Round\_8u\_Sfs, [2343](#)
- signal\_dot\_product
  - nppsDotProd\_16s16sc32fc, [2551](#)
  - nppsDotProd\_16s16sc32sc\_Sfs, [2552](#)
  - nppsDotProd\_16s16sc64sc, [2552](#)
  - nppsDotProd\_16s16sc\_Sfs, [2552](#)
  - nppsDotProd\_16s32f, [2553](#)
  - nppsDotProd\_16s32s32s\_Sfs, [2553](#)
  - nppsDotProd\_16s32s\_Sfs, [2554](#)
  - nppsDotProd\_16s64s, [2554](#)
  - nppsDotProd\_16s\_Sfs, [2554](#)
  - nppsDotProd\_16sc32fc, [2555](#)
  - nppsDotProd\_16sc32sc\_Sfs, [2555](#)
  - nppsDotProd\_16sc64sc, [2556](#)
  - nppsDotProd\_16sc\_Sfs, [2556](#)
  - nppsDotProd\_32f, [2556](#)
  - nppsDotProd\_32f32fc, [2557](#)
  - nppsDotProd\_32f32fc64fc, [2557](#)
  - nppsDotProd\_32f64f, [2557](#)
  - nppsDotProd\_32fc, [2558](#)
  - nppsDotProd\_32fc64fc, [2558](#)
  - nppsDotProd\_32s32sc\_Sfs, [2558](#)
  - nppsDotProd\_32s\_Sfs, [2559](#)
  - nppsDotProd\_32sc\_Sfs, [2559](#)
  - nppsDotProd\_64f, [2560](#)
  - nppsDotProd\_64f64fc, [2560](#)
  - nppsDotProd\_64fc, [2560](#)
  - nppsDotProdGetBufferSize\_16s16sc32fc, [2561](#)
  - nppsDotProdGetBufferSize\_16s16sc32sc\_Sfs, [2561](#)
  - nppsDotProdGetBufferSize\_16s16sc64sc, [2561](#)
  - nppsDotProdGetBufferSize\_16s16sc\_Sfs, [2562](#)
  - nppsDotProdGetBufferSize\_16s32f, [2562](#)
  - nppsDotProdGetBufferSize\_16s32s32s\_Sfs, [2562](#)
  - nppsDotProdGetBufferSize\_16s32s\_Sfs, [2562](#)
  - nppsDotProdGetBufferSize\_16s64s, [2563](#)
  - nppsDotProdGetBufferSize\_16s\_Sfs, [2563](#)
  - nppsDotProdGetBufferSize\_16sc32fc, [2563](#)
  - nppsDotProdGetBufferSize\_16sc32sc\_Sfs, [2563](#)
  - nppsDotProdGetBufferSize\_16sc64sc, [2564](#)
  - nppsDotProdGetBufferSize\_16sc\_Sfs, [2564](#)
  - nppsDotProdGetBufferSize\_32f, [2564](#)
  - nppsDotProdGetBufferSize\_32f32fc, [2564](#)
  - nppsDotProdGetBufferSize\_32f32fc64fc, [2565](#)
  - nppsDotProdGetBufferSize\_32f64f, [2565](#)
  - nppsDotProdGetBufferSize\_32fc, [2565](#)
  - nppsDotProdGetBufferSize\_32fc64fc, [2565](#)
  - nppsDotProdGetBufferSize\_32s32sc\_Sfs, [2566](#)
  - nppsDotProdGetBufferSize\_32s\_Sfs, [2566](#)
  - nppsDotProdGetBufferSize\_32sc\_Sfs, [2566](#)
  - nppsDotProdGetBufferSize\_64f, [2566](#)
  - nppsDotProdGetBufferSize\_64f64fc, [2567](#)
  - nppsDotProdGetBufferSize\_64fc, [2567](#)
- signal\_exp
  - nppsExp\_16s\_ISfs, [2362](#)
  - nppsExp\_16s\_Sfs, [2363](#)
  - nppsExp\_32f, [2363](#)
  - nppsExp\_32f64f, [2363](#)
  - nppsExp\_32f\_I, [2363](#)
  - nppsExp\_32s\_ISfs, [2364](#)

- nppsExp\_32s\_Sfs, [2364](#)
- nppsExp\_64f, [2364](#)
- nppsExp\_64f\_I, [2365](#)
- nppsExp\_64s\_ISfs, [2365](#)
- nppsExp\_64s\_Sfs, [2365](#)
- signal\_free
  - nppsFree, [2623](#)
- signal\_infinity\_norm
  - nppsNorm\_Inf\_16s32f, [2515](#)
  - nppsNorm\_Inf\_16s32s\_Sfs, [2515](#)
  - nppsNorm\_Inf\_32f, [2515](#)
  - nppsNorm\_Inf\_32fc32f, [2515](#)
  - nppsNorm\_Inf\_64f, [2516](#)
  - nppsNorm\_Inf\_64fc64f, [2516](#)
  - nppsNormInfGetBufferSize\_16s32f, [2516](#)
  - nppsNormInfGetBufferSize\_16s32s\_Sfs, [2517](#)
  - nppsNormInfGetBufferSize\_32f, [2517](#)
  - nppsNormInfGetBufferSize\_32fc32f, [2517](#)
  - nppsNormInfGetBufferSize\_64f, [2517](#)
  - nppsNormInfGetBufferSize\_64fc64f, [2518](#)
- signal\_infinity\_norm\_diff
  - nppsNormDiff\_Inf\_16s32f, [2532](#)
  - nppsNormDiff\_Inf\_16s32s\_Sfs, [2532](#)
  - nppsNormDiff\_Inf\_32f, [2532](#)
  - nppsNormDiff\_Inf\_32fc32f, [2533](#)
  - nppsNormDiff\_Inf\_64f, [2533](#)
  - nppsNormDiff\_Inf\_64fc64f, [2533](#)
  - nppsNormDiffInfGetBufferSize\_16s32f, [2534](#)
  - nppsNormDiffInfGetBufferSize\_16s32s\_Sfs, [2534](#)
  - nppsNormDiffInfGetBufferSize\_32f, [2534](#)
  - nppsNormDiffInfGetBufferSize\_32fc32f, [2535](#)
  - nppsNormDiffInfGetBufferSize\_64f, [2535](#)
  - nppsNormDiffInfGetBufferSize\_64fc64f, [2535](#)
- signal\_integral
  - nppsIntegral\_32s, [2442](#)
  - nppsIntegralGetBufferSize\_32s, [2442](#)
- signal\_inversetan
  - nppsArctan\_32f, [2375](#)
  - nppsArctan\_32f\_I, [2375](#)
  - nppsArctan\_64f, [2375](#)
  - nppsArctan\_64f\_I, [2376](#)
- signal\_L1\_norm
  - nppsNorm\_L1\_16s32f, [2520](#)
  - nppsNorm\_L1\_16s32s\_Sfs, [2520](#)
  - nppsNorm\_L1\_16s64s\_Sfs, [2520](#)
  - nppsNorm\_L1\_32f, [2521](#)
  - nppsNorm\_L1\_32fc64f, [2521](#)
  - nppsNorm\_L1\_64f, [2521](#)
  - nppsNorm\_L1\_64fc64f, [2522](#)
  - nppsNormL1GetBufferSize\_16s32f, [2522](#)
  - nppsNormL1GetBufferSize\_16s32s\_Sfs, [2522](#)
  - nppsNormL1GetBufferSize\_16s64s\_Sfs, [2522](#)
  - nppsNormL1GetBufferSize\_32f, [2523](#)
  - nppsNormL1GetBufferSize\_32fc64f, [2523](#)
  - nppsNormL1GetBufferSize\_64f, [2523](#)
  - nppsNormL1GetBufferSize\_64fc64f, [2523](#)
- signal\_L1\_norm\_diff
  - nppsNormDiff\_L1\_16s32f, [2537](#)
  - nppsNormDiff\_L1\_16s32s\_Sfs, [2537](#)
  - nppsNormDiff\_L1\_16s64s\_Sfs, [2537](#)
  - nppsNormDiff\_L1\_32f, [2538](#)
  - nppsNormDiff\_L1\_32fc64f, [2538](#)
  - nppsNormDiff\_L1\_64f, [2538](#)
  - nppsNormDiff\_L1\_64fc64f, [2539](#)
  - nppsNormDiffL1GetBufferSize\_16s32f, [2539](#)
  - nppsNormDiffL1GetBufferSize\_16s32s\_Sfs, [2539](#)
  - nppsNormDiffL1GetBufferSize\_16s64s\_Sfs, [2540](#)
  - nppsNormDiffL1GetBufferSize\_32f, [2540](#)
  - nppsNormDiffL1GetBufferSize\_32fc64f, [2540](#)
  - nppsNormDiffL1GetBufferSize\_64f, [2540](#)
  - nppsNormDiffL1GetBufferSize\_64fc64f, [2541](#)
- signal\_L2\_norm
  - nppsNorm\_L2\_16s32f, [2526](#)
  - nppsNorm\_L2\_16s32s\_Sfs, [2526](#)
  - nppsNorm\_L2\_32f, [2526](#)
  - nppsNorm\_L2\_32fc64f, [2527](#)
  - nppsNorm\_L2\_64f, [2527](#)
  - nppsNorm\_L2\_64fc64f, [2527](#)
  - nppsNorm\_L2Sqr\_16s64s\_Sfs, [2528](#)
  - nppsNormL2GetBufferSize\_16s32f, [2528](#)
  - nppsNormL2GetBufferSize\_16s32s\_Sfs, [2528](#)
  - nppsNormL2GetBufferSize\_32f, [2528](#)
  - nppsNormL2GetBufferSize\_32fc64f, [2529](#)
  - nppsNormL2GetBufferSize\_64f, [2529](#)
  - nppsNormL2GetBufferSize\_64fc64f, [2529](#)
  - nppsNormL2SqrGetBufferSize\_16s64s\_Sfs, [2529](#)
- signal\_L2\_norm\_diff
  - nppsNormDiff\_L2\_16s32f, [2543](#)
  - nppsNormDiff\_L2\_16s32s\_Sfs, [2543](#)
  - nppsNormDiff\_L2\_32f, [2543](#)
  - nppsNormDiff\_L2\_32fc64f, [2544](#)
  - nppsNormDiff\_L2\_64f, [2544](#)
  - nppsNormDiff\_L2\_64fc64f, [2544](#)
  - nppsNormDiff\_L2Sqr\_16s64s\_Sfs, [2545](#)
  - nppsNormDiffL2GetBufferSize\_16s32f, [2545](#)
  - nppsNormDiffL2GetBufferSize\_16s32s\_Sfs, [2545](#)
  - nppsNormDiffL2GetBufferSize\_32f, [2546](#)
  - nppsNormDiffL2GetBufferSize\_32fc64f, [2546](#)

- [nppsNormDiffL2GetBufferSize\\_64f, 2546](#)
  - [nppsNormDiffL2GetBufferSize\\_64fc64f, 2546](#)
  - [nppsNormDiffL2SqrGetBufferSize\\_16s64s\\_Sfs, 2547](#)
- [signal\\_ln](#)
  - [nppsLn\\_16s\\_ISfs, 2366](#)
  - [nppsLn\\_16s\\_Sfs, 2367](#)
  - [nppsLn\\_32f, 2367](#)
  - [nppsLn\\_32f\\_I, 2367](#)
  - [nppsLn\\_32s16s\\_Sfs, 2367](#)
  - [nppsLn\\_32s\\_ISfs, 2368](#)
  - [nppsLn\\_32s\\_Sfs, 2368](#)
  - [nppsLn\\_64f, 2368](#)
  - [nppsLn\\_64f32f, 2369](#)
  - [nppsLn\\_64f\\_I, 2369](#)
- [signal\\_lshifc](#)
  - [nppsLShiftC\\_16s, 2404](#)
  - [nppsLShiftC\\_16s\\_I, 2405](#)
  - [nppsLShiftC\\_16u, 2405](#)
  - [nppsLShiftC\\_16u\\_I, 2405](#)
  - [nppsLShiftC\\_32s, 2405](#)
  - [nppsLShiftC\\_32s\\_I, 2406](#)
  - [nppsLShiftC\\_32u, 2406](#)
  - [nppsLShiftC\\_32u\\_I, 2406](#)
  - [nppsLShiftC\\_8u, 2407](#)
  - [nppsLShiftC\\_8u\\_I, 2407](#)
- [signal\\_malloc](#)
  - [nppsMalloc\\_16s, 2619](#)
  - [nppsMalloc\\_16sc, 2619](#)
  - [nppsMalloc\\_16u, 2619](#)
  - [nppsMalloc\\_32f, 2619](#)
  - [nppsMalloc\\_32fc, 2620](#)
  - [nppsMalloc\\_32s, 2620](#)
  - [nppsMalloc\\_32sc, 2620](#)
  - [nppsMalloc\\_32u, 2620](#)
  - [nppsMalloc\\_64f, 2621](#)
  - [nppsMalloc\\_64fc, 2621](#)
  - [nppsMalloc\\_64s, 2621](#)
  - [nppsMalloc\\_64sc, 2621](#)
  - [nppsMalloc\\_8s, 2622](#)
  - [nppsMalloc\\_8u, 2622](#)
- [signal\\_max](#)
  - [nppsMax\\_16s, 2470](#)
  - [nppsMax\\_32f, 2471](#)
  - [nppsMax\\_32s, 2471](#)
  - [nppsMax\\_64f, 2471](#)
  - [nppsMaxAbs\\_16s, 2472](#)
  - [nppsMaxAbs\\_32s, 2472](#)
  - [nppsMaxAbsGetBufferSize\\_16s, 2472](#)
  - [nppsMaxAbsGetBufferSize\\_32s, 2473](#)
  - [nppsMaxAbsIdx\\_16s, 2473](#)
  - [nppsMaxAbsIdx\\_32s, 2473](#)
  - [nppsMaxAbsIdxGetBufferSize\\_16s, 2474](#)
  - [nppsMaxAbsIdxGetBufferSize\\_32s, 2474](#)
  - [nppsMaxGetBufferSize\\_16s, 2474](#)
  - [nppsMaxGetBufferSize\\_32f, 2474](#)
  - [nppsMaxGetBufferSize\\_32s, 2475](#)
  - [nppsMaxGetBufferSize\\_64f, 2475](#)
  - [nppsMaxIdx\\_16s, 2475](#)
  - [nppsMaxIdx\\_32f, 2476](#)
  - [nppsMaxIdx\\_32s, 2476](#)
  - [nppsMaxIdx\\_64f, 2476](#)
  - [nppsMaxIdxGetBufferSize\\_16s, 2477](#)
  - [nppsMaxIdxGetBufferSize\\_32f, 2477](#)
  - [nppsMaxIdxGetBufferSize\\_32s, 2477](#)
  - [nppsMaxIdxGetBufferSize\\_64f, 2478](#)
- [signal\\_maximum\\_error](#)
  - [nppsMaximumError\\_16s, 2573](#)
  - [nppsMaximumError\\_16sc, 2573](#)
  - [nppsMaximumError\\_16u, 2573](#)
  - [nppsMaximumError\\_32f, 2574](#)
  - [nppsMaximumError\\_32fc, 2574](#)
  - [nppsMaximumError\\_32s, 2574](#)
  - [nppsMaximumError\\_32sc, 2575](#)
  - [nppsMaximumError\\_32u, 2575](#)
  - [nppsMaximumError\\_64f, 2575](#)
  - [nppsMaximumError\\_64fc, 2576](#)
  - [nppsMaximumError\\_64s, 2576](#)
  - [nppsMaximumError\\_64sc, 2576](#)
  - [nppsMaximumError\\_8s, 2577](#)
  - [nppsMaximumError\\_8u, 2577](#)
  - [nppsMaximumErrorGetBufferSize\\_16s, 2577](#)
  - [nppsMaximumErrorGetBufferSize\\_16sc, 2578](#)
  - [nppsMaximumErrorGetBufferSize\\_16u, 2578](#)
  - [nppsMaximumErrorGetBufferSize\\_32f, 2578](#)
  - [nppsMaximumErrorGetBufferSize\\_32fc, 2578](#)
  - [nppsMaximumErrorGetBufferSize\\_32s, 2579](#)
  - [nppsMaximumErrorGetBufferSize\\_32sc, 2579](#)
  - [nppsMaximumErrorGetBufferSize\\_32u, 2579](#)
  - [nppsMaximumErrorGetBufferSize\\_64f, 2579](#)
  - [nppsMaximumErrorGetBufferSize\\_64fc, 2580](#)
  - [nppsMaximumErrorGetBufferSize\\_64s, 2580](#)
  - [nppsMaximumErrorGetBufferSize\\_64sc, 2580](#)
  - [nppsMaximumErrorGetBufferSize\\_8s, 2580](#)
  - [nppsMaximumErrorGetBufferSize\\_8u, 2581](#)
- [signal\\_maximum\\_relative\\_error](#)
  - [nppsMaximumRelativeError\\_16s, 2595](#)
  - [nppsMaximumRelativeError\\_16sc, 2595](#)
  - [nppsMaximumRelativeError\\_16u, 2596](#)
  - [nppsMaximumRelativeError\\_32f, 2596](#)
  - [nppsMaximumRelativeError\\_32fc, 2596](#)
  - [nppsMaximumRelativeError\\_32s, 2597](#)
  - [nppsMaximumRelativeError\\_32sc, 2597](#)
  - [nppsMaximumRelativeError\\_32u, 2598](#)
  - [nppsMaximumRelativeError\\_64f, 2598](#)
  - [nppsMaximumRelativeError\\_64fc, 2598](#)
  - [nppsMaximumRelativeError\\_64s, 2599](#)

- nppsMaximumRelativeError\_64sc, 2599
- nppsMaximumRelativeError\_8s, 2600
- nppsMaximumRelativeError\_8u, 2600
- nppsMaximumRelativeErrorGetBufferSize\_-16s, 2600
- nppsMaximumRelativeErrorGetBufferSize\_-16sc, 2601
- nppsMaximumRelativeErrorGetBufferSize\_-16u, 2601
- nppsMaximumRelativeErrorGetBufferSize\_-32f, 2601
- nppsMaximumRelativeErrorGetBufferSize\_-32fc, 2601
- nppsMaximumRelativeErrorGetBufferSize\_-32s, 2602
- nppsMaximumRelativeErrorGetBufferSize\_-32sc, 2602
- nppsMaximumRelativeErrorGetBufferSize\_-32u, 2602
- nppsMaximumRelativeErrorGetBufferSize\_-64f, 2602
- nppsMaximumRelativeErrorGetBufferSize\_-64fc, 2603
- nppsMaximumRelativeErrorGetBufferSize\_-64s, 2603
- nppsMaximumRelativeErrorGetBufferSize\_-64sc, 2603
- nppsMaximumRelativeErrorGetBufferSize\_-8s, 2603
- nppsMaximumRelativeErrorGetBufferSize\_-8u, 2604
- signal\_mean
  - nppsMean\_16s\_Sfs, 2490
  - nppsMean\_16sc\_Sfs, 2490
  - nppsMean\_32f, 2490
  - nppsMean\_32fc, 2491
  - nppsMean\_32s\_Sfs, 2491
  - nppsMean\_64f, 2491
  - nppsMean\_64fc, 2492
  - nppsMeanGetBufferSize\_16s\_Sfs, 2492
  - nppsMeanGetBufferSize\_16sc\_Sfs, 2492
  - nppsMeanGetBufferSize\_32f, 2493
  - nppsMeanGetBufferSize\_32fc, 2493
  - nppsMeanGetBufferSize\_32s\_Sfs, 2493
  - nppsMeanGetBufferSize\_64f, 2493
  - nppsMeanGetBufferSize\_64fc, 2494
- signal\_mean\_and\_standard\_deviation
  - nppsMeanStdDev\_16s32s\_Sfs, 2498
  - nppsMeanStdDev\_16s\_Sfs, 2499
  - nppsMeanStdDev\_32f, 2499
  - nppsMeanStdDev\_64f, 2499
  - nppsMeanStdDevGetBufferSize\_16s32s\_Sfs, 2500
  - nppsMeanStdDevGetBufferSize\_16s\_Sfs, 2500
  - nppsMeanStdDevGetBufferSize\_32f, 2500
  - nppsMeanStdDevGetBufferSize\_64f, 2500
- signal\_min
  - nppsMin\_16s, 2480
  - nppsMin\_32f, 2481
  - nppsMin\_32s, 2481
  - nppsMin\_64f, 2481
  - nppsMinAbs\_16s, 2482
  - nppsMinAbs\_32s, 2482
  - nppsMinAbsGetBufferSize\_16s, 2482
  - nppsMinAbsGetBufferSize\_32s, 2483
  - nppsMinAbsIdx\_16s, 2483
  - nppsMinAbsIdx\_32s, 2483
  - nppsMinAbsIdxGetBufferSize\_16s, 2484
  - nppsMinAbsIdxGetBufferSize\_32s, 2484
  - nppsMinGetBufferSize\_16s, 2484
  - nppsMinGetBufferSize\_32f, 2484
  - nppsMinGetBufferSize\_32s, 2485
  - nppsMinGetBufferSize\_64f, 2485
  - nppsMinIdx\_16s, 2485
  - nppsMinIdx\_32f, 2486
  - nppsMinIdx\_32s, 2486
  - nppsMinIdx\_64f, 2486
  - nppsMinIdxGetBufferSize\_16s, 2487
  - nppsMinIdxGetBufferSize\_32f, 2487
  - nppsMinIdxGetBufferSize\_32s, 2487
  - nppsMinIdxGetBufferSize\_64f, 2488
- signal\_min\_every\_or\_max\_every
  - nppsMaxEvery\_16s\_I, 2458
  - nppsMaxEvery\_16u\_I, 2459
  - nppsMaxEvery\_32f\_I, 2459
  - nppsMaxEvery\_32s\_I, 2459
  - nppsMaxEvery\_8u\_I, 2459
  - nppsMinEvery\_16s\_I, 2460
  - nppsMinEvery\_16u\_I, 2460
  - nppsMinEvery\_32f\_I, 2460
  - nppsMinEvery\_32s\_I, 2461
  - nppsMinEvery\_64f\_I, 2461
  - nppsMinEvery\_8u\_I, 2461
- signal\_min\_max
  - nppsMinMax\_16s, 2504
  - nppsMinMax\_16u, 2504
  - nppsMinMax\_32f, 2504
  - nppsMinMax\_32s, 2505
  - nppsMinMax\_32u, 2505
  - nppsMinMax\_64f, 2505
  - nppsMinMax\_8u, 2506
  - nppsMinMaxGetBufferSize\_16s, 2506
  - nppsMinMaxGetBufferSize\_16u, 2506
  - nppsMinMaxGetBufferSize\_32f, 2507
  - nppsMinMaxGetBufferSize\_32s, 2507
  - nppsMinMaxGetBufferSize\_32u, 2507

- [nppsMinMaxGetBufferSize\\_64f, 2507](#)
  - [nppsMinMaxGetBufferSize\\_8u, 2508](#)
  - [nppsMinMaxIndx\\_16s, 2508](#)
  - [nppsMinMaxIndx\\_16u, 2508](#)
  - [nppsMinMaxIndx\\_32f, 2509](#)
  - [nppsMinMaxIndx\\_32s, 2509](#)
  - [nppsMinMaxIndx\\_32u, 2510](#)
  - [nppsMinMaxIndx\\_64f, 2510](#)
  - [nppsMinMaxIndx\\_8u, 2510](#)
  - [nppsMinMaxIndxGetBufferSize\\_16s, 2511](#)
  - [nppsMinMaxIndxGetBufferSize\\_16u, 2511](#)
  - [nppsMinMaxIndxGetBufferSize\\_32f, 2511](#)
  - [nppsMinMaxIndxGetBufferSize\\_32s, 2512](#)
  - [nppsMinMaxIndxGetBufferSize\\_32u, 2512](#)
  - [nppsMinMaxIndxGetBufferSize\\_64f, 2512](#)
  - [nppsMinMaxIndxGetBufferSize\\_8u, 2512](#)
- [signal\\_mul](#)
  - [nppsMul\\_16s, 2312](#)
  - [nppsMul\\_16s32f, 2312](#)
  - [nppsMul\\_16s32s\\_Sfs, 2313](#)
  - [nppsMul\\_16s\\_I, 2313](#)
  - [nppsMul\\_16s\\_ISfs, 2313](#)
  - [nppsMul\\_16s\\_Sfs, 2314](#)
  - [nppsMul\\_16sc\\_ISfs, 2314](#)
  - [nppsMul\\_16sc\\_Sfs, 2314](#)
  - [nppsMul\\_16u16s\\_Sfs, 2315](#)
  - [nppsMul\\_16u\\_ISfs, 2315](#)
  - [nppsMul\\_16u\\_Sfs, 2315](#)
  - [nppsMul\\_32f, 2316](#)
  - [nppsMul\\_32f32fc, 2316](#)
  - [nppsMul\\_32f32fc\\_I, 2316](#)
  - [nppsMul\\_32f\\_I, 2317](#)
  - [nppsMul\\_32fc, 2317](#)
  - [nppsMul\\_32fc\\_I, 2317](#)
  - [nppsMul\\_32s32sc\\_ISfs, 2318](#)
  - [nppsMul\\_32s32sc\\_Sfs, 2318](#)
  - [nppsMul\\_32s\\_ISfs, 2318](#)
  - [nppsMul\\_32s\\_Sfs, 2319](#)
  - [nppsMul\\_32sc\\_ISfs, 2319](#)
  - [nppsMul\\_32sc\\_Sfs, 2319](#)
  - [nppsMul\\_64f, 2320](#)
  - [nppsMul\\_64f\\_I, 2320](#)
  - [nppsMul\\_64fc, 2320](#)
  - [nppsMul\\_64fc\\_I, 2321](#)
  - [nppsMul\\_8u16u, 2321](#)
  - [nppsMul\\_8u\\_ISfs, 2321](#)
  - [nppsMul\\_8u\\_Sfs, 2322](#)
  - [nppsMul\\_Low\\_32s\\_Sfs, 2322](#)
- [signal\\_mulc](#)
  - [nppsMulC\\_16s\\_ISfs, 2258](#)
  - [nppsMulC\\_16s\\_Sfs, 2259](#)
  - [nppsMulC\\_16sc\\_ISfs, 2259](#)
  - [nppsMulC\\_16sc\\_Sfs, 2259](#)
  - [nppsMulC\\_16u\\_ISfs, 2260](#)
  - [nppsMulC\\_16u\\_Sfs, 2260](#)
  - [nppsMulC\\_32f, 2260](#)
  - [nppsMulC\\_32f16s\\_Sfs, 2261](#)
  - [nppsMulC\\_32f\\_I, 2261](#)
  - [nppsMulC\\_32fc, 2261](#)
  - [nppsMulC\\_32fc\\_I, 2262](#)
  - [nppsMulC\\_32s\\_ISfs, 2262](#)
  - [nppsMulC\\_32s\\_Sfs, 2262](#)
  - [nppsMulC\\_32sc\\_ISfs, 2263](#)
  - [nppsMulC\\_32sc\\_Sfs, 2263](#)
  - [nppsMulC\\_64f, 2263](#)
  - [nppsMulC\\_64f64s\\_ISfs, 2264](#)
  - [nppsMulC\\_64f\\_I, 2264](#)
  - [nppsMulC\\_64fc, 2264](#)
  - [nppsMulC\\_64fc\\_I, 2265](#)
  - [nppsMulC\\_8u\\_ISfs, 2265](#)
  - [nppsMulC\\_8u\\_Sfs, 2265](#)
  - [nppsMulC\\_Low\\_32f16s, 2266](#)
- [signal\\_normalize](#)
  - [nppsNormalize\\_16s\\_Sfs, 2377](#)
  - [nppsNormalize\\_16sc\\_Sfs, 2378](#)
  - [nppsNormalize\\_32f, 2378](#)
  - [nppsNormalize\\_32fc, 2378](#)
  - [nppsNormalize\\_64f, 2379](#)
  - [nppsNormalize\\_64fc, 2379](#)
- [signal\\_not](#)
  - [nppsNot\\_16u, 2401](#)
  - [nppsNot\\_16u\\_I, 2401](#)
  - [nppsNot\\_32u, 2402](#)
  - [nppsNot\\_32u\\_I, 2402](#)
  - [nppsNot\\_8u, 2402](#)
  - [nppsNot\\_8u\\_I, 2402](#)
- [signal\\_or](#)
  - [nppsOr\\_16u, 2392](#)
  - [nppsOr\\_16u\\_I, 2392](#)
  - [nppsOr\\_32u, 2393](#)
  - [nppsOr\\_32u\\_I, 2393](#)
  - [nppsOr\\_8u, 2393](#)
  - [nppsOr\\_8u\\_I, 2394](#)
- [signal\\_orc](#)
  - [nppsOrC\\_16u, 2389](#)
  - [nppsOrC\\_16u\\_I, 2389](#)
  - [nppsOrC\\_32u, 2390](#)
  - [nppsOrC\\_32u\\_I, 2390](#)
  - [nppsOrC\\_8u, 2390](#)
  - [nppsOrC\\_8u\\_I, 2391](#)
- [signal\\_rshifc](#)
  - [nppsRShiftC\\_16s, 2408](#)
  - [nppsRShiftC\\_16s\\_I, 2409](#)
  - [nppsRShiftC\\_16u, 2409](#)
  - [nppsRShiftC\\_16u\\_I, 2409](#)
  - [nppsRShiftC\\_32s, 2409](#)
  - [nppsRShiftC\\_32s\\_I, 2410](#)
  - [nppsRShiftC\\_32u, 2410](#)



- nppsRShiftC\_32u\_I, [2410](#)
- nppsRShiftC\_8u, [2411](#)
- nppsRShiftC\_8u\_I, [2411](#)
- signal\_set
  - nppsSet\_16s, [2445](#)
  - nppsSet\_16sc, [2445](#)
  - nppsSet\_16u, [2445](#)
  - nppsSet\_32f, [2445](#)
  - nppsSet\_32fc, [2446](#)
  - nppsSet\_32s, [2446](#)
  - nppsSet\_32sc, [2446](#)
  - nppsSet\_32u, [2446](#)
  - nppsSet\_64f, [2447](#)
  - nppsSet\_64fc, [2447](#)
  - nppsSet\_64s, [2447](#)
  - nppsSet\_64sc, [2448](#)
  - nppsSet\_8s, [2448](#)
  - nppsSet\_8u, [2448](#)
- signal\_sqrt
  - nppsSqrt\_16s\_ISfs, [2354](#)
  - nppsSqrt\_16s\_Sfs, [2354](#)
  - nppsSqrt\_16sc\_ISfs, [2355](#)
  - nppsSqrt\_16sc\_Sfs, [2355](#)
  - nppsSqrt\_16u\_ISfs, [2355](#)
  - nppsSqrt\_16u\_Sfs, [2355](#)
  - nppsSqrt\_32f, [2356](#)
  - nppsSqrt\_32f\_I, [2356](#)
  - nppsSqrt\_32fc, [2356](#)
  - nppsSqrt\_32fc\_I, [2357](#)
  - nppsSqrt\_32s16s\_Sfs, [2357](#)
  - nppsSqrt\_64f, [2357](#)
  - nppsSqrt\_64f\_I, [2357](#)
  - nppsSqrt\_64fc, [2358](#)
  - nppsSqrt\_64fc\_I, [2358](#)
  - nppsSqrt\_64s16s\_Sfs, [2358](#)
  - nppsSqrt\_64s\_ISfs, [2358](#)
  - nppsSqrt\_64s\_Sfs, [2359](#)
  - nppsSqrt\_8u\_ISfs, [2359](#)
  - nppsSqrt\_8u\_Sfs, [2359](#)
- signal\_square
  - nppsSqr\_16s\_ISfs, [2348](#)
  - nppsSqr\_16s\_Sfs, [2348](#)
  - nppsSqr\_16sc\_ISfs, [2348](#)
  - nppsSqr\_16sc\_Sfs, [2349](#)
  - nppsSqr\_16u\_ISfs, [2349](#)
  - nppsSqr\_16u\_Sfs, [2349](#)
  - nppsSqr\_32f, [2349](#)
  - nppsSqr\_32f\_I, [2350](#)
  - nppsSqr\_32fc, [2350](#)
  - nppsSqr\_32fc\_I, [2350](#)
  - nppsSqr\_64f, [2350](#)
  - nppsSqr\_64f\_I, [2351](#)
  - nppsSqr\_64fc, [2351](#)
  - nppsSqr\_64fc\_I, [2351](#)
  - nppsSqr\_8u\_ISfs, [2351](#)
  - nppsSqr\_8u\_Sfs, [2351](#)
- signal\_standard\_deviation
  - nppsStdDev\_16s32s\_Sfs, [2495](#)
  - nppsStdDev\_16s\_Sfs, [2495](#)
  - nppsStdDev\_32f, [2496](#)
  - nppsStdDev\_64f, [2496](#)
  - nppsStdDevGetBufferSize\_16s32s\_Sfs, [2496](#)
  - nppsStdDevGetBufferSize\_16s\_Sfs, [2497](#)
  - nppsStdDevGetBufferSize\_32f, [2497](#)
  - nppsStdDevGetBufferSize\_64f, [2497](#)
- signal\_sub
  - nppsSub\_16s, [2324](#)
  - nppsSub\_16s32f, [2325](#)
  - nppsSub\_16s\_I, [2325](#)
  - nppsSub\_16s\_ISfs, [2325](#)
  - nppsSub\_16s\_Sfs, [2326](#)
  - nppsSub\_16sc\_ISfs, [2326](#)
  - nppsSub\_16sc\_Sfs, [2326](#)
  - nppsSub\_16u\_ISfs, [2327](#)
  - nppsSub\_16u\_Sfs, [2327](#)
  - nppsSub\_32f, [2327](#)
  - nppsSub\_32f\_I, [2328](#)
  - nppsSub\_32fc, [2328](#)
  - nppsSub\_32fc\_I, [2328](#)
  - nppsSub\_32s\_ISfs, [2328](#)
  - nppsSub\_32s\_Sfs, [2329](#)
  - nppsSub\_32sc\_ISfs, [2329](#)
  - nppsSub\_32sc\_Sfs, [2329](#)
  - nppsSub\_64f, [2330](#)
  - nppsSub\_64f\_I, [2330](#)
  - nppsSub\_64fc, [2330](#)
  - nppsSub\_64fc\_I, [2331](#)
  - nppsSub\_8u\_ISfs, [2331](#)
  - nppsSub\_8u\_Sfs, [2331](#)
- signal\_subc
  - nppsSubC\_16s\_ISfs, [2268](#)
  - nppsSubC\_16s\_Sfs, [2268](#)
  - nppsSubC\_16sc\_ISfs, [2269](#)
  - nppsSubC\_16sc\_Sfs, [2269](#)
  - nppsSubC\_16u\_ISfs, [2269](#)
  - nppsSubC\_16u\_Sfs, [2270](#)
  - nppsSubC\_32f, [2270](#)
  - nppsSubC\_32f\_I, [2270](#)
  - nppsSubC\_32fc, [2271](#)
  - nppsSubC\_32fc\_I, [2271](#)
  - nppsSubC\_32s\_ISfs, [2271](#)
  - nppsSubC\_32s\_Sfs, [2272](#)
  - nppsSubC\_32sc\_ISfs, [2272](#)
  - nppsSubC\_32sc\_Sfs, [2272](#)
  - nppsSubC\_64f, [2273](#)
  - nppsSubC\_64f\_I, [2273](#)
  - nppsSubC\_64fc, [2273](#)
  - nppsSubC\_64fc\_I, [2274](#)

- [nppsSubC\\_8u\\_ISfs, 2274](#)
  - [nppsSubC\\_8u\\_Sfs, 2274](#)
- [signal\\_subcrev](#)
  - [nppsSubCRev\\_16s\\_ISfs, 2277](#)
  - [nppsSubCRev\\_16s\\_Sfs, 2278](#)
  - [nppsSubCRev\\_16sc\\_ISfs, 2278](#)
  - [nppsSubCRev\\_16sc\\_Sfs, 2278](#)
  - [nppsSubCRev\\_16u\\_ISfs, 2279](#)
  - [nppsSubCRev\\_16u\\_Sfs, 2279](#)
  - [nppsSubCRev\\_32f, 2279](#)
  - [nppsSubCRev\\_32f\\_I, 2280](#)
  - [nppsSubCRev\\_32fc, 2280](#)
  - [nppsSubCRev\\_32fc\\_I, 2280](#)
  - [nppsSubCRev\\_32s\\_ISfs, 2280](#)
  - [nppsSubCRev\\_32s\\_Sfs, 2281](#)
  - [nppsSubCRev\\_32sc\\_ISfs, 2281](#)
  - [nppsSubCRev\\_32sc\\_Sfs, 2281](#)
  - [nppsSubCRev\\_64f, 2282](#)
  - [nppsSubCRev\\_64f\\_I, 2282](#)
  - [nppsSubCRev\\_64fc, 2282](#)
  - [nppsSubCRev\\_64fc\\_I, 2283](#)
  - [nppsSubCRev\\_8u\\_ISfs, 2283](#)
  - [nppsSubCRev\\_8u\\_Sfs, 2283](#)
- [signal\\_sum](#)
  - [nppsSum\\_16s32s\\_Sfs, 2463](#)
  - [nppsSum\\_16s\\_Sfs, 2463](#)
  - [nppsSum\\_16sc32sc\\_Sfs, 2464](#)
  - [nppsSum\\_16sc\\_Sfs, 2464](#)
  - [nppsSum\\_32f, 2464](#)
  - [nppsSum\\_32fc, 2465](#)
  - [nppsSum\\_32s\\_Sfs, 2465](#)
  - [nppsSum\\_64f, 2465](#)
  - [nppsSum\\_64fc, 2466](#)
  - [nppsSumGetBufferSize\\_16s32s\\_Sfs, 2466](#)
  - [nppsSumGetBufferSize\\_16s\\_Sfs, 2466](#)
  - [nppsSumGetBufferSize\\_16sc32sc\\_Sfs, 2467](#)
  - [nppsSumGetBufferSize\\_16sc\\_Sfs, 2467](#)
  - [nppsSumGetBufferSize\\_32f, 2467](#)
  - [nppsSumGetBufferSize\\_32fc, 2467](#)
  - [nppsSumGetBufferSize\\_32s\\_Sfs, 2468](#)
  - [nppsSumGetBufferSize\\_64f, 2468](#)
  - [nppsSumGetBufferSize\\_64fc, 2468](#)
- [signal\\_sumln](#)
  - [nppsSumLn\\_16s32f, 2371](#)
  - [nppsSumLn\\_32f, 2372](#)
  - [nppsSumLn\\_32f64f, 2372](#)
  - [nppsSumLn\\_64f, 2372](#)
  - [nppsSumLnGetBufferSize\\_16s32f, 2373](#)
  - [nppsSumLnGetBufferSize\\_32f, 2373](#)
  - [nppsSumLnGetBufferSize\\_32f64f, 2373](#)
  - [nppsSumLnGetBufferSize\\_64f, 2373](#)
- [signal\\_threshold](#)
  - [nppsThreshold\\_16s, 2420](#)
  - [nppsThreshold\\_16s\\_I, 2421](#)
  - [nppsThreshold\\_16sc, 2421](#)
  - [nppsThreshold\\_16sc\\_I, 2421](#)
  - [nppsThreshold\\_32f, 2422](#)
  - [nppsThreshold\\_32f\\_I, 2422](#)
  - [nppsThreshold\\_32fc, 2422](#)
  - [nppsThreshold\\_32fc\\_I, 2423](#)
  - [nppsThreshold\\_64f, 2423](#)
  - [nppsThreshold\\_64f\\_I, 2423](#)
  - [nppsThreshold\\_64fc, 2424](#)
  - [nppsThreshold\\_64fc\\_I, 2424](#)
  - [nppsThreshold\\_GT\\_16s, 2424](#)
  - [nppsThreshold\\_GT\\_16s\\_I, 2425](#)
  - [nppsThreshold\\_GT\\_16sc, 2425](#)
  - [nppsThreshold\\_GT\\_16sc\\_I, 2425](#)
  - [nppsThreshold\\_GT\\_32f, 2426](#)
  - [nppsThreshold\\_GT\\_32f\\_I, 2426](#)
  - [nppsThreshold\\_GT\\_32fc, 2426](#)
  - [nppsThreshold\\_GT\\_32fc\\_I, 2427](#)
  - [nppsThreshold\\_GT\\_64f, 2427](#)
  - [nppsThreshold\\_GT\\_64f\\_I, 2427](#)
  - [nppsThreshold\\_GT\\_64fc, 2428](#)
  - [nppsThreshold\\_GT\\_64fc\\_I, 2428](#)
  - [nppsThreshold\\_GTV\\_16s, 2428](#)
  - [nppsThreshold\\_GTV\\_16s\\_I, 2429](#)
  - [nppsThreshold\\_GTV\\_16sc, 2429](#)
  - [nppsThreshold\\_GTV\\_16sc\\_I, 2429](#)
  - [nppsThreshold\\_GTV\\_32f, 2430](#)
  - [nppsThreshold\\_GTV\\_32f\\_I, 2430](#)
  - [nppsThreshold\\_GTV\\_32fc, 2430](#)
  - [nppsThreshold\\_GTV\\_32fc\\_I, 2431](#)
  - [nppsThreshold\\_GTV\\_64f, 2431](#)
  - [nppsThreshold\\_GTV\\_64f\\_I, 2431](#)
  - [nppsThreshold\\_GTV\\_64fc, 2432](#)
  - [nppsThreshold\\_GTV\\_64fc\\_I, 2432](#)
  - [nppsThreshold\\_LT\\_16s, 2432](#)
  - [nppsThreshold\\_LT\\_16s\\_I, 2433](#)
  - [nppsThreshold\\_LT\\_16sc, 2433](#)
  - [nppsThreshold\\_LT\\_16sc\\_I, 2433](#)
  - [nppsThreshold\\_LT\\_32f, 2434](#)
  - [nppsThreshold\\_LT\\_32f\\_I, 2434](#)
  - [nppsThreshold\\_LT\\_32fc, 2434](#)
  - [nppsThreshold\\_LT\\_32fc\\_I, 2435](#)
  - [nppsThreshold\\_LT\\_64f, 2435](#)
  - [nppsThreshold\\_LT\\_64f\\_I, 2435](#)
  - [nppsThreshold\\_LT\\_64fc, 2436](#)
  - [nppsThreshold\\_LT\\_64fc\\_I, 2436](#)
  - [nppsThreshold\\_LTV\\_16s, 2436](#)
  - [nppsThreshold\\_LTV\\_16s\\_I, 2437](#)
  - [nppsThreshold\\_LTV\\_16sc, 2437](#)
  - [nppsThreshold\\_LTV\\_16sc\\_I, 2437](#)
  - [nppsThreshold\\_LTV\\_32f, 2438](#)
  - [nppsThreshold\\_LTV\\_32f\\_I, 2438](#)
  - [nppsThreshold\\_LTV\\_32fc, 2438](#)
  - [nppsThreshold\\_LTV\\_32fc\\_I, 2439](#)

- nppsThreshold\_LTVal\_64f, [2439](#)
- nppsThreshold\_LTVal\_64f\_I, [2439](#)
- nppsThreshold\_LTVal\_64fc, [2440](#)
- nppsThreshold\_LTVal\_64fc\_I, [2440](#)
- signal\_xor
  - nppsXor\_16u, [2398](#)
  - nppsXor\_16u\_I, [2398](#)
  - nppsXor\_32u, [2399](#)
  - nppsXor\_32u\_I, [2399](#)
  - nppsXor\_8u, [2399](#)
  - nppsXor\_8u\_I, [2400](#)
- signal\_xorc
  - nppsXorC\_16u, [2395](#)
  - nppsXorC\_16u\_I, [2395](#)
  - nppsXorC\_32u, [2396](#)
  - nppsXorC\_32u\_I, [2396](#)
  - nppsXorC\_8u, [2396](#)
  - nppsXorC\_8u\_I, [2397](#)
- signal\_zero
  - nppsZero\_16s, [2449](#)
  - nppsZero\_16sc, [2450](#)
  - nppsZero\_32f, [2450](#)
  - nppsZero\_32fc, [2450](#)
  - nppsZero\_32s, [2450](#)
  - nppsZero\_32sc, [2450](#)
  - nppsZero\_64f, [2451](#)
  - nppsZero\_64fc, [2451](#)
  - nppsZero\_64s, [2451](#)
  - nppsZero\_64sc, [2451](#)
  - nppsZero\_8u, [2452](#)
- Sqr, [330](#), [2347](#)
- SqrDistanceFull\_Norm, [1886](#)
- sqrdistancefullnorm
  - nppiSqrDistanceFull\_Norm\_16u32f\_AC4R, [1888](#)
  - nppiSqrDistanceFull\_Norm\_16u32f\_C1R, [1888](#)
  - nppiSqrDistanceFull\_Norm\_16u32f\_C3R, [1888](#)
  - nppiSqrDistanceFull\_Norm\_16u32f\_C4R, [1889](#)
  - nppiSqrDistanceFull\_Norm\_32f\_AC4R, [1889](#)
  - nppiSqrDistanceFull\_Norm\_32f\_C1R, [1890](#)
  - nppiSqrDistanceFull\_Norm\_32f\_C3R, [1890](#)
  - nppiSqrDistanceFull\_Norm\_32f\_C4R, [1891](#)
  - nppiSqrDistanceFull\_Norm\_8s32f\_AC4R, [1891](#)
  - nppiSqrDistanceFull\_Norm\_8s32f\_C1R, [1891](#)
  - nppiSqrDistanceFull\_Norm\_8s32f\_C3R, [1892](#)
  - nppiSqrDistanceFull\_Norm\_8s32f\_C4R, [1892](#)
  - nppiSqrDistanceFull\_Norm\_8u32f\_AC4R, [1893](#)
  - nppiSqrDistanceFull\_Norm\_8u32f\_C1R, [1893](#)
  - nppiSqrDistanceFull\_Norm\_8u32f\_C3R, [1894](#)
- nppiSqrDistanceFull\_Norm\_8u32f\_C4R, [1894](#)
- nppiSqrDistanceFull\_Norm\_8u\_AC4RSfs, [1894](#)
- nppiSqrDistanceFull\_Norm\_8u\_C1RSfs, [1895](#)
- nppiSqrDistanceFull\_Norm\_8u\_C3RSfs, [1895](#)
- nppiSqrDistanceFull\_Norm\_8u\_C4RSfs, [1896](#)
- SqrDistanceSame\_Norm, [1897](#)
- sqrdistancesamenorm
  - nppiSqrDistanceSame\_Norm\_16u32f\_AC4R, [1899](#)
  - nppiSqrDistanceSame\_Norm\_16u32f\_C1R, [1899](#)
  - nppiSqrDistanceSame\_Norm\_16u32f\_C3R, [1900](#)
  - nppiSqrDistanceSame\_Norm\_16u32f\_C4R, [1900](#)
  - nppiSqrDistanceSame\_Norm\_32f\_AC4R, [1900](#)
  - nppiSqrDistanceSame\_Norm\_32f\_C1R, [1901](#)
  - nppiSqrDistanceSame\_Norm\_32f\_C3R, [1901](#)
  - nppiSqrDistanceSame\_Norm\_32f\_C4R, [1902](#)
  - nppiSqrDistanceSame\_Norm\_8s32f\_AC4R, [1902](#)
  - nppiSqrDistanceSame\_Norm\_8s32f\_C1R, [1903](#)
  - nppiSqrDistanceSame\_Norm\_8s32f\_C3R, [1903](#)
  - nppiSqrDistanceSame\_Norm\_8s32f\_C4R, [1903](#)
  - nppiSqrDistanceSame\_Norm\_8u32f\_AC4R, [1904](#)
  - nppiSqrDistanceSame\_Norm\_8u32f\_C1R, [1904](#)
  - nppiSqrDistanceSame\_Norm\_8u32f\_C3R, [1905](#)
  - nppiSqrDistanceSame\_Norm\_8u32f\_C4R, [1905](#)
  - nppiSqrDistanceSame\_Norm\_8u\_AC4RSfs, [1906](#)
  - nppiSqrDistanceSame\_Norm\_8u\_C1RSfs, [1906](#)
  - nppiSqrDistanceSame\_Norm\_8u\_C3RSfs, [1907](#)
  - nppiSqrDistanceSame\_Norm\_8u\_C4RSfs, [1907](#)
- SqrDistanceValid\_Norm, [1908](#)
- sqrdistancevalidnorm
  - nppiSqrDistanceValid\_Norm\_16u32f\_AC4R, [1910](#)
  - nppiSqrDistanceValid\_Norm\_16u32f\_C1R, [1910](#)
  - nppiSqrDistanceValid\_Norm\_16u32f\_C3R, [1911](#)



- nppiSqrDistanceValid\_Norm\_16u32f\_C4R, 1911
- nppiSqrDistanceValid\_Norm\_32f\_AC4R, 1911
- nppiSqrDistanceValid\_Norm\_32f\_C1R, 1912
- nppiSqrDistanceValid\_Norm\_32f\_C3R, 1912
- nppiSqrDistanceValid\_Norm\_32f\_C4R, 1913
- nppiSqrDistanceValid\_Norm\_8s32f\_AC4R, 1913
- nppiSqrDistanceValid\_Norm\_8s32f\_C1R, 1914
- nppiSqrDistanceValid\_Norm\_8s32f\_C3R, 1914
- nppiSqrDistanceValid\_Norm\_8s32f\_C4R, 1914
- nppiSqrDistanceValid\_Norm\_8u32f\_AC4R, 1915
- nppiSqrDistanceValid\_Norm\_8u32f\_C1R, 1915
- nppiSqrDistanceValid\_Norm\_8u32f\_C3R, 1916
- nppiSqrDistanceValid\_Norm\_8u32f\_C4R, 1916
- nppiSqrDistanceValid\_Norm\_8u\_AC4RSfs, 1917
- nppiSqrDistanceValid\_Norm\_8u\_C1RSfs, 1917
- nppiSqrDistanceValid\_Norm\_8u\_C3RSfs, 1918
- nppiSqrDistanceValid\_Norm\_8u\_C4RSfs, 1918
- SqrIntegral, 1848
- Sqrt, 344, 2353
- Standard Deviation, 2495
- Statistical Functions, 2457
- Statistical Operations, 1393
- Sub, 246, 2323
- SubC, 114, 2267
- SubCRev, 2276
- Sum, 1460, 2462
- SumLn, 2371
- Swap Channels, 923
- Threshold, 2416
- Threshold and Compare Operations, 2130
- Threshold Operations, 2131
- Transpose, 916
- typedefs\_npp
  - NPP\_AFFINE\_QUAD\_INCORRECT\_WARNING, 46
  - NPP\_ALG\_HINT\_ACCURATE, 41
  - NPP\_ALG\_HINT\_FAST, 41
  - NPP\_ALG\_HINT\_NONE, 41
  - NPP\_ALIGNMENT\_ERROR, 44
  - NPP\_ANCHOR\_ERROR, 45
  - NPP\_BAD\_ARGUMENT\_ERROR, 45
  - NPP\_BORDER\_CONSTANT, 42
  - NPP\_BORDER\_NONE, 42
  - NPP\_BORDER\_REPLICATE, 42
  - NPP\_BORDER\_UNDEFINED, 42
  - NPP\_BORDER\_WRAP, 42
  - NPP\_BOTH\_AXIS, 42
  - NPP\_CHANNEL\_ERROR, 45
  - NPP\_CHANNEL\_ORDER\_ERROR, 45
  - NPP\_CMP\_EQ, 41
  - NPP\_CMP\_GREATER, 41
  - NPP\_CMP\_GREATER\_EQ, 41
  - NPP\_CMP\_LESS, 41
  - NPP\_CMP\_LESS\_EQ, 41
  - NPP\_COEFFICIENT\_ERROR, 45
  - NPP\_COI\_ERROR, 45
  - NPP\_CONTEXT\_MATCH\_ERROR, 45
  - NPP\_CUDA\_1\_0, 41
  - NPP\_CUDA\_1\_1, 41
  - NPP\_CUDA\_1\_2, 41
  - NPP\_CUDA\_1\_3, 41
  - NPP\_CUDA\_2\_0, 41
  - NPP\_CUDA\_2\_1, 41
  - NPP\_CUDA\_3\_0, 41
  - NPP\_CUDA\_3\_2, 41
  - NPP\_CUDA\_3\_5, 41
  - NPP\_CUDA\_5\_0, 41
  - NPP\_CUDA\_KERNEL\_EXECUTION\_ERROR, 44
  - NPP\_CUDA\_NOT\_CAPABLE, 41
  - NPP\_CUDA\_UNKNOWN\_VERSION, 41
  - NPP\_DATA\_TYPE\_ERROR, 45
  - NPP\_DIVIDE\_BY\_ZERO\_ERROR, 45
  - NPP\_DIVIDE\_BY\_ZERO\_WARNING, 46
  - NPP\_DIVISOR\_ERROR, 45
  - NPP\_DOUBLE\_SIZE\_WARNING, 46
  - NPP\_ERROR, 45
  - NPP\_ERROR\_RESERVED, 45
  - NPP\_FFT\_FLAG\_ERROR, 45
  - NPP\_FFT\_ORDER\_ERROR, 45
  - NPP\_HAAR\_CLASSIFIER\_PIXEL\_MATCH\_ERROR, 44
  - NPP\_HISTOGRAM\_NUMBER\_OF\_LEVELS\_ERROR, 45
  - NPP\_HORIZONTAL\_AXIS, 42
  - NPP\_INTERPOLATION\_ERROR, 45
  - NPP\_INVALID\_DEVICE\_POINTER\_ERROR, 44
  - NPP\_INVALID\_HOST\_POINTER\_ERROR, 44
  - NPP\_LUT\_NUMBER\_OF\_LEVELS\_ERROR, 45
  - NPP\_LUT\_PALETTE\_BITSIZE\_ERROR, 44

- NPP\_MASK\_SIZE\_1\_X\_3, 43
- NPP\_MASK\_SIZE\_1\_X\_5, 43
- NPP\_MASK\_SIZE\_3\_X\_1, 43
- NPP\_MASK\_SIZE\_3\_X\_3, 43
- NPP\_MASK\_SIZE\_5\_X\_1, 43
- NPP\_MASK\_SIZE\_5\_X\_5, 43
- NPP\_MASK\_SIZE\_ERROR, 45
- NPP\_MEMCPY\_ERROR, 44
- NPP\_MEMFREE\_ERROR, 44
- NPP\_MEMORY\_ALLOCATION\_ERR, 45
- NPP\_MEMSET\_ERROR, 44
- NPP\_MIRROR\_FLIP\_ERROR, 45
- NPP\_MISALIGNED\_DST\_ROI\_WARNING, 46
- NPP\_MOMENT\_00\_ZERO\_ERROR, 45
- NPP\_NO\_ERROR, 45
- NPP\_NO\_MEMORY\_ERROR, 45
- NPP\_NO\_OPERATION\_WARNING, 46
- NPP\_NOT\_EVEN\_STEP\_ERROR, 45
- NPP\_NOT\_IMPLEMENTED\_ERROR, 45
- NPP\_NOT\_SUFFICIENT\_COMPUTE\_CAPABILITY, 44
- NPP\_NOT\_SUPPORTED\_MODE\_ERROR, 44
- NPP\_NULL\_POINTER\_ERROR, 45
- NPP\_NUMBER\_OF\_CHANNELS\_ERROR, 45
- NPP\_OUT\_OFF\_RANGE\_ERROR, 45
- NPP\_QUADRANGLE\_ERROR, 45
- NPP\_QUALITY\_INDEX\_ERROR, 44
- NPP\_RANGE\_ERROR, 45
- NPP\_RECTANGLE\_ERROR, 45
- NPP\_RESIZE\_FACTOR\_ERROR, 45
- NPP\_RESIZE\_NO\_OPERATION\_ERROR, 44
- NPP\_RND\_FINANCIAL, 44
- NPP\_RND\_NEAR, 43
- NPP\_RND\_ZERO, 44
- NPP\_ROUND\_MODE\_NOT\_SUPPORTED\_ERROR, 44
- NPP\_ROUND\_NEAREST\_TIES\_AWAY\_FROM\_ZERO, 44
- NPP\_ROUND\_NEAREST\_TIES\_TO\_EVEN, 44
- NPP\_ROUND\_TOWARD\_ZERO, 44
- NPP\_SCALE\_RANGE\_ERROR, 45
- NPP\_SIZE\_ERROR, 45
- NPP\_STEP\_ERROR, 45
- NPP\_STRIDE\_ERROR, 45
- NPP\_SUCCESS, 46
- NPP\_TEXTURE\_BIND\_ERROR, 44
- NPP\_THRESHOLD\_ERROR, 45
- NPP\_THRESHOLD\_NEGATIVE\_LEVEL\_ERROR, 45
- NPP\_VERTICAL\_AXIS, 42
- NPP\_WRONG\_INTERSECTION\_QUAD\_WARNING, 46
- NPP\_WRONG\_INTERSECTION\_ROI\_ERROR, 44
- NPP\_WRONG\_INTERSECTION\_ROI\_WARNING, 46
- NPP\_ZC\_MODE\_NOT\_SUPPORTED\_ERROR, 44
- NPP\_ZERO\_MASK\_VALUE\_ERROR, 45
- NPPI\_INTER\_CUBIC, 43
- NPPI\_INTER\_CUBIC2P\_B05C03, 43
- NPPI\_INTER\_CUBIC2P\_BSPLINE, 43
- NPPI\_INTER\_CUBIC2P\_CATMULLROM, 43
- NPPI\_INTER\_LANCZOS, 43
- NPPI\_INTER\_LINEAR, 43
- NPPI\_INTER\_NN, 43
- NPPI\_INTER\_SUPER, 43
- NPPI\_INTER\_UNDEFINED, 43
- NPPI\_OP\_ALPHA\_ATOP, 42
- NPPI\_OP\_ALPHA\_ATOP\_PREMUL, 42
- NPPI\_OP\_ALPHA\_IN, 42
- NPPI\_OP\_ALPHA\_IN\_PREMUL, 42
- NPPI\_OP\_ALPHA\_OUT, 42
- NPPI\_OP\_ALPHA\_OUT\_PREMUL, 42
- NPPI\_OP\_ALPHA\_OVER, 42
- NPPI\_OP\_ALPHA\_OVER\_PREMUL, 42
- NPPI\_OP\_ALPHA\_PLUS, 42
- NPPI\_OP\_ALPHA\_PLUS\_PREMUL, 42
- NPPI\_OP\_ALPHA\_PREMUL, 42
- NPPI\_OP\_ALPHA\_XOR, 42
- NPPI\_OP\_ALPHA\_XOR\_PREMUL, 42
- NPPI\_SMOOTH\_EDGE, 43
- npPiACTable, 42
- npPiDCTable, 42
- nppZCC, 46
- nppZCR, 46
- nppZCXor, 46
- typedefs\_npp
  - NPP\_MAX\_16S, 39
  - NPP\_MAX\_16U, 39
  - NPP\_MAX\_32S, 39
  - NPP\_MAX\_32U, 39
  - NPP\_MAX\_64S, 39
  - NPP\_MAX\_64U, 39
  - NPP\_MAX\_8S, 39
  - NPP\_MAX\_8U, 40
  - NPP\_MAXABS\_32F, 40
  - NPP\_MAXABS\_64F, 40
  - NPP\_MIN\_16S, 40
  - NPP\_MIN\_16U, 40
  - NPP\_MIN\_32S, 40
  - NPP\_MIN\_32U, 40

- NPP\_MIN\_64S, [40](#)
- NPP\_MIN\_64U, [40](#)
- NPP\_MIN\_8S, [40](#)
- NPP\_MIN\_8U, [40](#)
- NPP\_MINABS\_32F, [41](#)
- NPP\_MINABS\_64F, [41](#)
- NppCmpOp, [41](#)
- NppGpuComputeCapability, [41](#)
- NppHintAlgorithm, [41](#)
- NppiAlphaOp, [41](#)
- NppiAxis, [42](#)
- NppiBorderType, [42](#)
- NppiHuffmanTableType, [42](#)
- NppiInterpolationMode, [42](#)
- NppiMaskSize, [43](#)
- NppRoundMode, [43](#)
- NppStatus, [44](#)
- NppsZCType, [46](#)
- width
  - NppiRect, [2632](#)
  - NppiSize, [2633](#)
- x
  - NppiPoint, [2631](#)
  - NppiRect, [2632](#)
- Xor, [456](#), [2398](#)
- XorC, [393](#), [2395](#)
- y
  - NppiPoint, [2631](#)
  - NppiRect, [2632](#)
- Zero, [2449](#)