Technical Brief

Mipmapping Normal Maps

DEVELOPMENT

Table of Contents

Mipmapping Normal Maps

by Michael Toksvig

Abstract

The result of averaging or interpolating unit normals is less than unit length unless all normals within the footprint happen to be identical. Most algorithms simply renormalize, but this paper explores how the shortening can be used as a measure of normal variation to eliminate the common problem of strobing/sparkling noise due to aliasing of specular highlights.

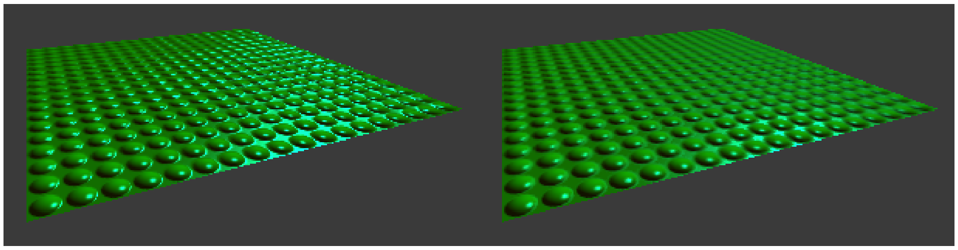

Figure 1. With aliasing (left) and without (right)

A very inexpensive implementation that simply substitutes a power function with a 2D texture lookup is presented. Other applications of the technique, such as free gloss mapping, are also discussed.

Estimating Normal Variation

Clearly the shortened normal does not encapsulate all information about how normals vary within the footprint.

An average normal with length $\sqrt{\frac{3}{4}}$ could indicate a highly anisotropic distribution with two groups of normals, each 30 degrees from the average, in opposite directions. Or the distribution could be completely isotropic with normals arranged in a cone

A good middle ground is to assume a Gaussian distribution of the angular deviation, α with a standard deviation of σ. Figure 1 shows the length of the average normal as a function of the standard deviation:

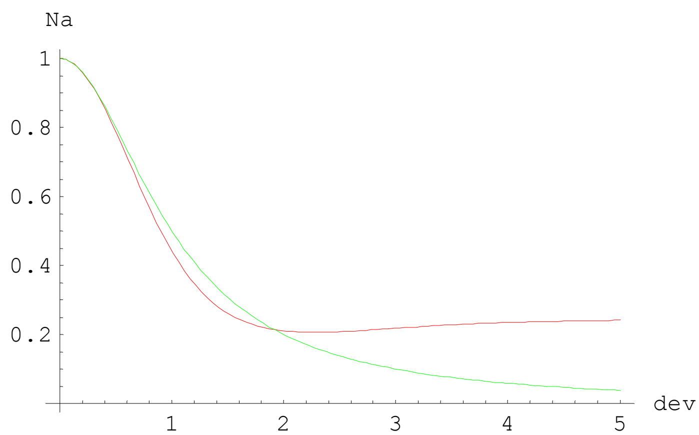

Figure 2. $|N_a|$ as Function of Standard Deviation, (Red), and Approximation (Green)

It can be seen that a Gaussian distribution will never produce a very short average normal, whereas in practice, arbitrarily short average normals may be observed in extreme cases. We therefore use the following approximation (order σ^3 Taylor approximation of reciprocal) with more desirable asymptotic behavior (green in Figure 2):

$$|N_a| = \frac{1}{1+\sigma^2} \Leftrightarrow \sigma^2 = \frac{1-|N_a|}{|N_a|} \tag{1}$$

So given a normal that has been shortened by averaging across a normal-map footprint, we can readily estimate the standard deviation, σ, of the contributing normals using the above formula

Eliminating Aliasing of Specular Highlights

One way to exploit this information is to eliminate the aliasing of specular highlights on bump-mapped surfaces, which is often quite objectionable. This problem has been addressed using less affordable techniques, e.g. in [Fournier 92] and [Shilling 97].

Traditionally, specular highlights are calculated using the renormalized average normal, e.g. like so like so for Blinn-Phong[1]:

$$\left(\frac{N_a \cdot H}{|N_a|}\right)^s, \tag{2}$$

where H is the half-angle vector, and s is the shininess exponent [Blinn 77].

[1] The technique is trivially applicable to Phong and other lighting models as well

Ideally, however, you would want to sum the power function for each normal, N_i , in the normal-map footprint, like so:

$$\sum_{footprint} (N_i \cdot H)^s \tag{3}$$

Since the power function does not combine linearly, the approximation in (2) deteriorates as the variation among the normals increases

A better approximation can be obtained by combining the bell curve from the shininess exponent, s, with the bell curve from the Gaussian distribution of normals.

Convolving the two functions directly would expand the support to encompass the entire sphere (and is quite an unsightly expression). Instead we approximate the combined functions with a new power function.

The shininess exponent, s', for the combined bell curves is derived by approximating both curves with Gaussian distributions on a plane.

To approximate the power function with a Gaussian, we use:

$$\cos\alpha \approx e^{-\frac{\alpha^2}{2}} \text{ and therefore } \cos^s \alpha \approx e^{-\frac{\alpha^2}{2}s} = e^{-\frac{\alpha^2}{2\sigma_s^2}} \text{, where } \sigma_s = s^{-\frac{1}{2}} \tag{4}$$

As can be seen from the radial plot in Figure 3, this approximation is pretty good, especially as s grows:

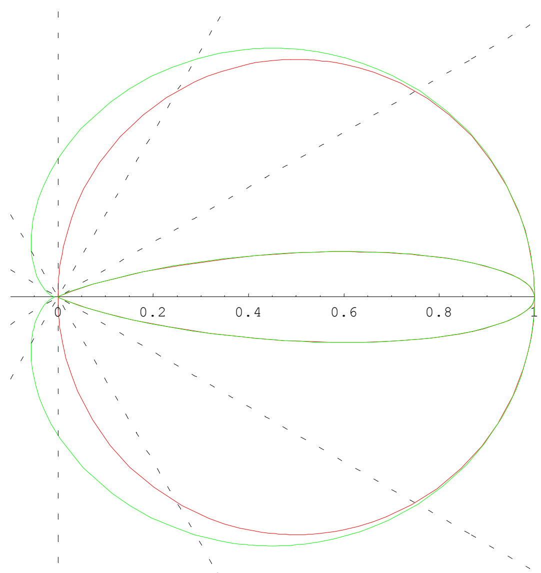

Figure 3. Radial Plot of Cosine (red) and Gaussian (green) for $s = 1$ (wide) and $s = 40$ (narrow lobe)

It is well known that convolving two Gaussians on a plane with standard deviations σ and σ_s will yield another Gaussian with a larger standard deviation $\sigma_{s'}$ that can be derived from $\sigma_{s'}^2 = \sigma^2 + \sigma_s^2$. Using (4) twice, we have:

$$\frac{1}{s'} = \sigma^2 + \frac{1}{s} \Leftrightarrow s' = \frac{1}{1 + s\,\sigma^2}\, s \tag{5}$$

For convenience, we introduce a "Toksvig factor", f_t, the factor by which the original shininess exponent, s, is to be scaled down:

$$f_t = \frac{1}{1 + s\,\sigma^2} = \frac{|N_a|}{|N_a| + s(1 - |N_a|)} \tag{6}$$

To preserve energy, the intensity of the widened highlight needs to be scaled down, too, so using:

$$\int_{hemisphere} \cos^s \alpha \, d\alpha = \frac{2\pi}{s+1} \tag{7}$$

we find the following expression for the specular component:

$$\frac{1 + f_t s}{1 + s} \left(\frac{N_a \cdot H}{|N_a|} \right)^{f_t s}, \tag{8}$$

which is just the Blinn-Phong expression with a lower exponent and a scale factor.

If the length of the average normal, $|N_a|$ is low (the normals vary a lot), the Toksvig factor, f_t, will be close to 0. This makes the specular highlight wider and fainter making the surface appear more dull. Conversely, if $|N_a|$ is close to 1 (the normals within the footprint are pretty much in agreement), f_t is close to 1 and the surface is unchanged.

Implementation

In hardware, traditional specular exponentiation is often accomplished using a table lookup in a 1D texture (one table per exponent).

In pseudo code, this looks roughly like:

```
Look up N_a in normal-map
Renormalize: N = N_a / |N_a|
Compute N·H
Look up result in 1D table using N·H
```

To eliminate aliasing, replace with this code:

```
Look up N_a in normal-map
Compute N_a·H
Compute N_a·N_a
Look up result in 2D table using N_a·H and N_a·N_a
```

The 2D table texture contains the expression from (8) expressed in terms of $N_a \cdot H$ and $N_a \cdot N_a$:

$$\frac{1+f_t s}{1+s}\left(\frac{N_a \cdot H}{\sqrt{N_a \cdot N_a}}\right)^{f_t s} \quad \text{where} \quad f_t = \frac{\sqrt{N_a \cdot N_a}}{\sqrt{N_a \cdot N_a} + s\left(1-\sqrt{N_a \cdot N_a}\right)} \tag{9}$$

Note that the row corresponding to $N_a \cdot N_a = 1$ is identical to the 1D table texture, and that the row corresponding to $N_a \cdot N_a = 0$ is constant $\frac{1}{1+s}$.

Also note that $N_a \cdot H$ can never be greater than $\sqrt{N_a \cdot N_a}$, so 1/3 of the texture will never be accessed

Finally note that the intensity of the highlight is sensitive to the error in the length of the average normal. Figure 4 shows the relative intensity of the highlight as a function of the length of the average normal.

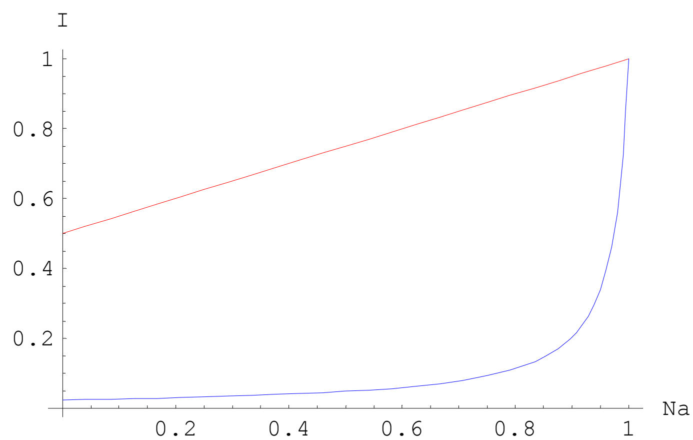

Figure 4. Relative Intensity of Highlight for s = 1 (red) and s = 40 (blue)

The derivative of the relative intensity with respect to $|N_a|$, i.e. the sensitivity to error in $|N_a|$ is:

$$\frac{s^2}{(1+s)\left(|N_a| + s\left(1-|N_a|\right)\right)^2} \tag{10}$$

For $|N_a| = 1$, this expression becomes $\frac{s^2}{1+s}$, which suggests that high specular exponents require high precision normal maps.

Alternatively, the effect can be phased out as $|N_a|$ approaches 1 where the signal-to-noise ratio goes down. This, too, can be encoded in the 2D table texture. For example, if the normals are represented using 8-bit signed components, the error in $|N_a|$ is $\frac{\sqrt{3}}{127 \cdot 2} \approx 0.0068$, and the error in the relative intensity is therefore 0.0068 times the expression in (10). In the areas where this error exceeds some threshold, the expression in (9) is replaced with:

$$\frac{threshold}{error} antialiased + \left(1 - \frac{threshold}{error}\right) aliased \tag{11}$$

where *antialiased* is the expression in (9) and *aliased* is the traditional formula

Free Gloss Maps

One thing to notice is that varying the shininess exponent across a surface comes for free when applying this technique: Simply putting shorter-than-unit normals in one area of the base normal-map will make that area appear duller. Using (6), it can be seen that a shininess of s, which must be smaller than the maximum shininess, s_0 is achieved when the normal in the base normal-map is scaled by:

$$\frac{s\,s_0}{s\,s_0 + s_0 - s} \tag{12}$$

Anti-aliasing Specular Highlights from Interpolated Normals

A variation of exploiting the shortening effect can also be applied to interpolated (i.e. not normal-mapped) normals, which show related aliasing artifacts. This technique involves creating a cube-map similar to a renormalization cube-map (each direction maps to a unit normal in that direction), but mipmapped without renormalization. Looking the interpolated normal up in this cube-map will produce a normal that has been shortened according to the rate of change in the original normal. This shortened normal is used in the same way as N_a above to make highlights wider and fainter when the normal varies greatly.

Environment Maps

Another variation can be applied when using environment maps (as suggested by Cass Everitt). When a ray is reflected in a normal mapped surface, the standard deviation of the normals within the footprint will be propagated to the reflection, thereby blurring the reflected image. This effect can be modeled by computing **`lod`** for the environment map as a function of $\log_2 \sigma$.

References

[Fournier 92] A. Fournier, Filtering Normal Maps and Creating Multiple Surfaces, Technical Report TR-92-41, Department of Computer Science, University of British Columbia, Vancouver, British Columbia, 1992

[Shilling 97] A. Schilling, Towards Real-Time Photorealistic Rendering: Challenges and Solutions, Proceedings of the 1997 SIGGRAPH /Eurographics Workshop on Graphics Hardware, pages 7-15, 1997

[Blinn 77] James F. Blinn, Models of Light Reflection for Computer Synthesized Pictures, Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques, p. 192-198, July 20-22, 1977, San Jose, California

[Dominé 03] Sébastien Dominé, Canonical Shaders for Optimal Performance, Australian Game Developers Conference 2003

Notice

ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, "MATERIALS") ARE BEING PROVIDED "AS IS." NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE.

Information furnished is believed to be accurate and reliable. However, NVIDIA Corporation assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of NVIDIA Corporation. Specifications mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. NVIDIA Corporation products are not authorized for use as critical components in life support devices or systems without express written approval of NVIDIA Corporation.

Trademarks

NVIDIA and the NVIDIA logo are trademarks or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

Copyright

NVIDIA Corporation
2701 San Tomas Expressway
Santa Clara, CA 95050
www.nvidia.com