Frequency Domain Normal Map Filtering

Abstract

Filtering is critical for representing image-based detail, such as textures or normal maps, across a variety of scales. While mipmapping textures is commonplace, accurate normal map filtering remains a challenging problem because of nonlinearities in shading—we cannot simply average nearby surface normals. In this paper, we show analytically that normal map filtering can be formalized as a spherical convolution of the *normal distribution function* (NDF) and the BRDF, for a large class of common BRDFs such as Lambertian, microfacet and factored measurements. This theoretical result explains many previous filtering techniques as special cases, and leads to a generalization to a broader class of measured and analytic BRDFs. Our practical algorithms leverage a significant body of previous work that has studied lighting-BRDF convolution. We show how spherical harmonics can be used to filter the NDF for Lambertian and low-frequency specular BRDFs, while spherical von Mises-Fisher distributions can be used for high-frequency materials.

1 Introduction

Representing image-based surface detail at a variety of scales requires good filtering algorithms. For texture mapping, aliasing is reduced by mipmapping [Williams 1983] or summed-area tables [Crow 1984]. Normal mapping (also known as bump mapping [Blinn 1978] or normal perturbation) is a simple and widely used analogue to texture mapping, that specifies the surface normal at each texel. Unfortunately, normal map filtering is very difficult because shading is not linear in the normal.

For example, consider the simple V-groove like structure or surface geometry in Fig. 1a. Initially, this spans two pixels, each of which has distinct normals (b). As we zoom out (c), the average normal of the two sides (e) corresponds simply to a flat surface, where the shading is likely to be very different. By contrast, our method preserves the full normal distribution (d), showing how to convolve it with the BRDF (f) to get an accurate result.

A more complex example is Fig. 2. Initially (top row), minimal filtering is required, and all methods perform identically. However, as we zoom out (middle and especially bottom rows), we quickly obtain radically different results, depending on if we use standard mipmap filtering or our method—which is close to ground truth.[1]

We develop a comprehensive framework for normal map filtering, in the context of real-time GPU rendering of normal maps, significantly generalizing the state of the art [Tan et al. 2005].

Theory of Filtering as Convolution of BRDF and NDF: Our most important contribution is theoretical. In Sec. 4, we derive an analytic formula, showing that filtering can be formally represented as a spherical convolution of the BRDF of the material, and the *normal distribution function*[2] or NDF for that texel. The NDF is a weighted mapping of surface normals onto the unit sphere; more formally, it is the extended Gaussian Image [Horn 1984] of the geometry within a texel. The theory applies equally to the conventional discrete normal maps shown in all our examples, as well as analytically-specified normal distributions.

This mathematical form holds for a large class of common BRDFs (including Lambertian, microfacet models and factored half-angle measurements), and immediately connects geometrical normal map filtering with the older lighting-BRDF convolution result for appearance [Basri and Jacobs 2001; Ramamoorthi and Hanrahan 2001b]. Our result also unifies many previous normal map filtering approaches, that can be viewed as special cases. Moreover, we can immediately apply a host of mathematical representations originally developed for lighting-BRDF convolution. In particular, we develop the following new practical algorithms:

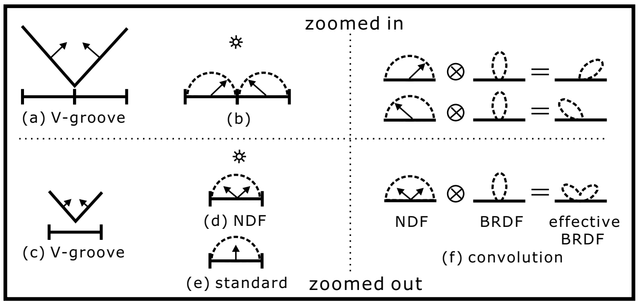


Figure 1: *Consider a simple V-groove. Initially (a), each face is a single pixel. As we zoom out, and average into a single pixel (c), standard mipmapping averages the normal to an effectively flat surface (e). However, our method computes a normal distribution function or NDF (d), that preserves the original normals. This NDF can be linearly convolved with the BRDF (right panel (f)) to obtain an effective BRDF, accurate for shading.*

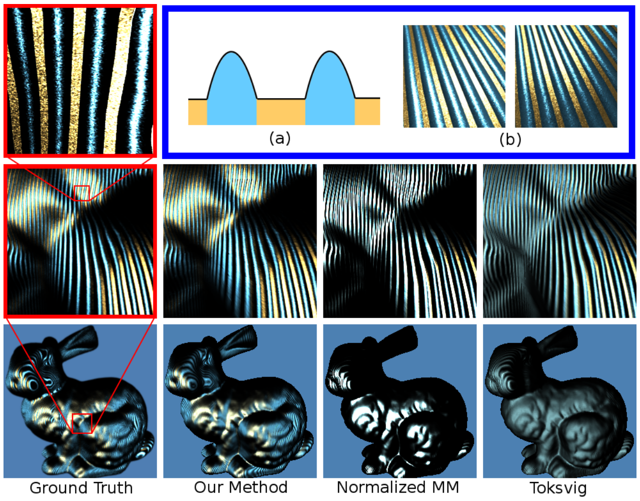


Figure 2: **Top:** *Closeups of the base normal map, which has different colors in flat regions and bumps to aid in comparison/visualization. All other methods are identical at this scale, and are not shown.* **Middle:** *When we zoom out, differences emerge between our (6-lobe) spherical vMF method, the Toksvig approach (rightmost), and a normalized mipmap. (Using unnormalized mipmap averaging of normals produces an essentially black image.)* **Bottom:** *Zooming out even further, our method is clearly more accurate than Toksvig's model (effectively single-lobe), and compares favorably with ground truth. (The reader may zoom into the PDF to compare images.)*

Spherical Harmonics: Spherical harmonic coefficients can be used to filter the NDF for Lambertian and low-frequency specular BRDFs (Sec. 5). We simply linearly average spherical harmonic coefficients at finer texels, as in mipmapping. To our knowledge, this is the first effective *linear* filtering algorithm for normal maps.

Spherical vMF Distributions: For rendering high-frequency materials, we use spherical expectation maximization [Banerjee et al. 2005] to fit von Mises-Fisher (vMF) spherical distributions [Fisher 1953] to the NDF, enabling a compact representation (Sec. 7). To our knowlege, this use of vMFs and EM for spherical distributions is new in graphics, and may have broader relevance.

Complex BRDFs and Lighting: Our framework allows us to handle complex BRDFs, including measured materials, as well as dynamically changing reflectance, lighting and view. We are also able to incorporate dynamic low-frequency environment maps.

[1] Ground truth is obtained by offline averaging or antialiasing of multiple (hundreds of) images created by jittering the camera, while rendering with normal maps at their finest resolution.

[2] Our normal distribution function is not to be confused with the similarly-named function from statistics.

2 Previous Work

The closest previous work is [Tan et al. 2005], which uses EM to fit Gaussian lobes at each texel. We introduce a new theoretical result in terms of the analysis of normal map filtering as convolution—it is easy to understand [Tan et al. 2005] as an important special case in this framework. Our formulation also allows spherical harmonic methods for low-frequency materials and Lambertian objects, which do not even require non-linear fitting of lobes. For high-frequency materials, by using vMFs and spherical EM, we use the natural spherical domain of surface normals—distorted planar or Euclidean projections, as in the planar Gaussian fits of [Tan et al. 2005], have been shown to considerably reduce accuracy, both in our work (Fig. 7), and in other contexts [Strehl et al. 2000]).

Note that [Tan et al. 2005] treat the BRDF itself as a pre-baked distribution of normals at fine-scale texels. We support and discuss this multi-scale tradeoff between BRDF and geometry (Sec. 6.2). However, our spherical representation enables us to derive a formal convolution result of the NDF with the BRDF, and allows us to separate or factor the two. The same NDF can be used with different, possibly non-Gaussian BRDFs, easily. We can also change the BRDF at runtime, and support dynamic complex lighting.

Normal Map Filtering: Earlier techniques based on lobe or peak-fitting include Schilling [1997] who uses a covariance matrix, and [Olano and North 1997] who map normal distributions consisting of a single 3D Gaussian. A simple recent GPU method is by Toksvig [2004]. In our framework, these techniques can retrospectively be considered similar to using a single vMF lobe. As seen in Figs. 2 and 7, a single lobe is not sufficient for complex NDFs.

An early inspiration is [Fournier 1992], who uses up to seven Phong peaks per texel, but needs up to 56 at the coarsest scales. As with [Tan et al. 2005], this is a special case of our framework, with some similarities to our spherical vMF algorithm. Note that [Fournier 1992] uses nonlinear least-squares optimization to fit peaks. In our experience, this is unstable and slow, especially considering the number of peaks and texels in a normal map.

Multi-Scale Representations: The idea of multiple scales, with bump or normal maps transitioning to BRDFs, is explored by [Becker and Max 1993], but they do not focus on normal map filtering as in our work. Similarly, appearance-preserving simplification methods replace fine-scale geometry with normal and texture maps [Cohen et al. 1998]. It is likely that our approach could enable continuous level of detail and antialiasing in these methods. Separately, our formulation allows one to understand the tradeoff between a normal distribution and the BRDF, since the final image is given by a convolution of the NDF and BRDF.

Displacement Maps: The power of modern GPUs has enabled more complex representations, like displacement maps with sphere tracing [Donnelly 2005]. A more data and compute-intensive approach is view-dependent displacement mapping [Wang et al. 2003]. However, these methods are more complicated, and antialiasing has not yet received significant attention, making filtered normal maps still the method of choice for many applications.

Convolution and Precomputed Radiance Transfer (PRT): Many of our mathematical representations derive from previous convolution and PRT methods. We utilize spherical harmonics [Ramamoorthi and Hanrahan 2001b; Sloan et al. 2002], including the 9 parameter formula for Lambertian reflectance [Ramamoorthi and Hanrahan 2001a; Basri and Jacobs 2001]. Our spherical vMF method extends zonal harmonic [Sloan et al. 2005] and spherical radial basis functions [Tsai and Shih 2006]. We also considered wavelet methods (introduced for reflectance in [Lalonde and Fournier 1997]), but found the number of terms for an artifact-free solution too large for practical use, even with smoother wavelets.[3]

We emphasize however, that ours is not a PRT algorithm, requiring minimal precomputation and working with conventional real-time rendering techniques. It is different from normal mapping in PRT [Sloan 2006], which presents a simple approximation and does not address filtering. Moreover, our method rests on an explicit analytic convolution formula, and uses the representations above solely for normal map filtering, not PRT.

3 Preliminaries

The reflected light B at a spatial point $\mathbf{x}$ is

$$B(\mathbf{x},\boldsymbol{\omega}_o) = \int_{S^2} L(\mathbf{x},\boldsymbol{\omega}_i)\rho(\boldsymbol{\omega}'_i,\boldsymbol{\omega}'_o)\,d\boldsymbol{\omega}_i, \tag{1}$$

where L is the lighting and ρ is the BRDF (actually, the transfer function including the cosine of the incident angle). L will generally be assumed to be from a small number of point lights (so equation 1 can be replaced by a summation over discrete $\boldsymbol{\omega}_i$). $\boldsymbol{\omega}'_i$ and $\boldsymbol{\omega}'_o$ are the local directions in the surface coordinate frame. To find them, we must rotate by R, that converts global directions ($\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_o$) to the local frame [Ramamoorthi and Hanrahan 2001b],

$$\boldsymbol{\omega}'_i = R(\mathbf{n})\boldsymbol{\omega}_i \qquad \boldsymbol{\omega}'_o = R(\mathbf{n})\boldsymbol{\omega}_o. \tag{2}$$

3.1 BRDF Representation and Parameterization

Effective BRDF: Therefore, we can consider an *effective BRDF* or transfer function that depends on the surface normal,

$$\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o,\mathbf{n}) = \rho(R(\mathbf{n})\boldsymbol{\omega}_i, R(\mathbf{n})\boldsymbol{\omega}_o), \tag{3}$$

with

$$B(\mathbf{x},\boldsymbol{\omega}_o) = \int_{S^2} L(\mathbf{x},\boldsymbol{\omega}_i)\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o,\mathbf{n}(\mathbf{x}))\,d\boldsymbol{\omega}_i. \tag{4}$$

BRDF Parameterizations: Many BRDFs can be written as

$$\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o,\mathbf{n}) = \rho(\boldsymbol{\omega}\cdot\mathbf{n}), \tag{5}$$

where the BRDF is a 1D function of a single variable or parameterization $\boldsymbol{\omega}$. In this paper, we focus most of our effort on these types of BRDFs, which encompass Lambertian, Blinn-Phong or microfacet half angle, and factored and measured BRDFs.

A very common example is Lambertian reflection, where the transfer function is simply the cosine of the incident angle, so that $\boldsymbol{\omega} = \boldsymbol{\omega}_i$, and $\rho = \max(\boldsymbol{\omega}\cdot\mathbf{n}, 0)$. The Blinn-Phong specular model with exponent s uses a transfer function of the form $\rho = (\boldsymbol{\omega}_h\cdot\mathbf{n})^s$, with the half-angle parameterization, $\boldsymbol{\omega} = \boldsymbol{\omega}_h$. Measured or procedural BRDF functions $\rho(\boldsymbol{\omega}_h\cdot\mathbf{n})$ can also be used.

A number of recent papers have proposed factored BRDFs for measured reflectance. [Lawrence et al. 2006] uses a factorization $f(\theta_h)g(\theta_d)$, in terms of half and difference angles. The $f(\theta_h)$ term clearly fits into the framework of equation 5, but the BRDF now also includes a product with $g(\theta_d)$. However, θ_d does not depend on $\mathbf{n}$ (and g does not need to be filtered), so we extend equation 5

$$\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o,\mathbf{n}) = \rho(\boldsymbol{\omega}\cdot\mathbf{n})g(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o), \tag{6}$$

where the g factor does not depend directly on $\mathbf{n}$.

3.2 Normal Map Representation and Filtering

Normal Map Input Representation: There are many equivalent normal map representations, including bump maps [Blinn 1978] and normal offsets. For simplicity, we use normal maps, parameterized on a plane, that directly specify the normal.

For rendering complex curved 3D geometry, the normal map $\mathbf{n}_j$ is rotated onto the surface using the geometric surface normal, i.e. $\hat{\mathbf{n}}_j = R(\hat{\mathbf{n}})\mathbf{n}_j$, where $\hat{\mathbf{n}}_j$ is the new normal, and $\hat{\mathbf{n}}$ is the geometric normal. The actual implementation is even simpler—we just perform all computations in the local frame of the geometric surface. Lighting and view are projected into this local frame, after which the planar normal map[4] is used directly, without explicit rotation.

[3] PRT methods can use a coarse wavelet approximation of the lighting, since it is not visualized directly, but we directly visualize NDF and BRDF.

[4] In practice, available normal maps do not usually use much higer resolutions than 512 × 512 or 1024 × 1024 for memory and practicality reasons. To obtain effectively higher or finer resolutions, we therefore often tile the base normal map multiple times over the surface. In rare cases, this introduces minor artifacts at tile boundaries that are unrelated to our algorithm.

For simplicity in the discussion below, the reader can therefore assume a planar underlying surface, while understanding that the extension to curved 3D geometry is straightforward.

Normal Map Filtering: Under standard linear filtering, coarser scales should represent the average radiance at the corresponding finer-level texels,

$$\begin{aligned} B(\mathbf{x},\boldsymbol{\omega}_o) &= \frac{1}{N}\sum_{j=1}^{N}\int_{S^2} L(\mathbf{x},\boldsymbol{\omega}_i)\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o,\mathbf{n}_j)\,d\boldsymbol{\omega}_i \\ &= \int_{S^2} L(\mathbf{x},\boldsymbol{\omega}_i)\left(\frac{1}{N}\sum_{j=1}^{N}\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o,\mathbf{n}_j)\right)d\boldsymbol{\omega}_i. \quad (7) \end{aligned}$$

This formulation allows us to define a new effective BRDF,

$$\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) = \frac{1}{N}\sum_{j=1}^{N}\rho(R(\mathbf{n}_j)\boldsymbol{\omega}_i, R(\mathbf{n}_j)\boldsymbol{\omega}_o). \qquad (8)$$

Once we have the effective BRDFs at each texel in the mipmap, we simply look up the effective BRDF at the appropriate scale, and use equation 4. However, explicitly tabulating an effective 4D BRDF at each texel involves very large storage costs. Moreover, the result depends on the BRDF ρ, which is often unknown a-priori.

This paper is about efficiently computing and representing ρ^{eff}. The next section shows how to explicitly factor ρ^{eff} as a convolution of the original BRDF and the NDF.

4 Theory of Normal Mapping as Convolution

In this section, we introduce our theoretical framework for normal map filtering as convolution. The next sections describe mathematical representations that can be used for practical implementation.

4.1 Normal Distribution Function (NDF)

Our first step is to convert equation 8 into continuous form, defining

$$\rho^{\text{eff}}(\boldsymbol{\omega}_i,\boldsymbol{\omega}_o) = \int_{S^2}\gamma(\mathbf{n})\rho(R(\mathbf{n})\boldsymbol{\omega}_i, R(\mathbf{n})\boldsymbol{\omega}_o)d\mathbf{n}, \qquad (9)$$

where $\gamma(\mathbf{n})$ is a normal distribution function, and the integral is over the sphere S^2 of surface orientations. For a discrete normal map, $\gamma(\mathbf{n})$ would simply be a sum of (spherical) delta distributions at $\mathbf{n}_j$, the fine-scale normals corresponding to that texel of the mipmap, i.e. $\gamma(\mathbf{n}) = \frac{1}{N}\sum_{j=1}^{N}\delta(\mathbf{n}-\mathbf{n}_j)$, as seen in Fig. 1d. For some procedurally generated normal maps, $\gamma(\mathbf{n})$ may be available analytically.

4.2 Frequency Domain Analysis in 2D

We now proceed to analyze equation 9 in the frequency domain. Many insights can be gained by starting in the simpler flatland or 2D case. In 2D, rotation simply corresponds to adding n,

$$\rho^{\text{eff}}(\omega_i,\omega_o) = \int_0^{2\pi}\gamma(n)\rho(\omega_i+n,\omega_o+n)\,dn. \qquad (10)$$

We can now consider expanding in Fourier series, by writing out $\gamma = \gamma_k F_k^*(n)$, and $\rho = \sum_{l,m}\rho_{lm}F_l(\omega_i+n)F_m(\omega_o+n)$, where $F_l(n)$ are the familiar Fourier basis functions $(2\pi)^{-1/2}\exp(\mathbf{i}ln)$, and F_k^* are the complex conjugates. Because $F_l(\omega+n) = \sqrt{2\pi}F_l(\omega)F_l(n)$,

$$\rho^{\text{eff}}(\omega_i,\omega_o) = 2\pi\sum_{k,l,m}\gamma_k\rho_{lm}F_l(\omega_i)F_m(\omega_o)\int_0^{2\pi}F_k^*(n)F_l(n)F_m(n)dn. \qquad (11)$$

We have grouped terms not depending on n away from those that do. The integral above involves a triple integral of Fourier series, and we denote the corresponding tripling coefficients C_{klm}. These tripling coefficients have recently been studied [Ng et al. 2004], and for Fourier series, they vanish unless $k = l+m$, when $C_{klm} = 1/\sqrt{2\pi}$. Noting that ρ^{eff} above is expressed in terms of $F_l(\omega_i)F_m(\omega_o)$, we can write a formula for its Fourier coefficients as

$$\rho^{\text{eff}}_{lm} = \sqrt{2\pi}\gamma_{l+m}\rho_{lm}. \qquad (12)$$

Discussion and Analogy with Convolution: This is a very simple product formula for the frequency coefficients of the effective BRDF, very similar to a convolution of the BRDF with the NDF. However, the convolution analogy is not exact, since equation 11 involves a triple integral (and n appears thrice in equation 10). In 3D, the formulae and sparsity for triple integals in the frequency domain (especially those involving rotations) are much more complicated [Ng et al. 2004]. Fortunately, many BRDFs are primarily functions of a single variable, $\rho(\boldsymbol{\omega}\cdot\mathbf{n})$ as in equation 5. In these cases, we will obtain a spherical convolution of the NDF and BRDF.

4.3 Frequency Domain Analysis in 3D

Plugging equation 5 into equation 9, we obtain

$$\rho^{\text{eff}}(\boldsymbol{\omega}) = \int_{S^2}\gamma(\mathbf{n})\rho(\boldsymbol{\omega}\cdot\mathbf{n})\,d\mathbf{n}. \qquad (13)$$

Note that the initial BRDF $\rho(\boldsymbol{\omega}\cdot\mathbf{n})$ is symmetric about $\mathbf{n}$, but the final result $\rho^{\text{eff}}(\boldsymbol{\omega})$ is a general function on the sphere.

Equation 13 expresses a spherical convolution of the NDF $\gamma(\mathbf{n})$ with the BRDF filter ρ. It is in fact exactly the same form of equation as derived for lighting-BRDF convolution (in the Lambertian or radially symmetric case) by [Basri and Jacobs 2001] and [Ramamoorthi and Hanrahan 2001b]. We simply have the NDF instead of the lighting. We now expand γ and ρ in spherical harmonics Y_{lm},

$$\gamma(\mathbf{n}) = \textstyle\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\gamma_{lm}Y_{lm}(\mathbf{n}) \qquad \rho(\boldsymbol{\omega}\cdot\mathbf{n}) = \displaystyle\sum_{l=0}^{\infty}\rho_l Y_{l0}(\boldsymbol{\omega}\cdot\mathbf{n}) \quad (14)$$

$$\rho^{\text{eff}}(\boldsymbol{\omega}) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l}\rho^{\text{eff}}_{lm}Y_{lm}(\boldsymbol{\omega}).$$

The convolution product formula in spherical harmonics is

$$\rho^{\text{eff}}_{lm} = \sqrt{\frac{4\pi}{2l+1}}\rho_l\gamma_{lm}. \qquad (15)$$

By defining $A_l = \sqrt{\frac{4\pi}{2l+1}}\rho_l$, we obtain

$$\boxed{\rho^{\text{eff}}_{lm}(q) = A_l\gamma_{lm}(q),} \qquad (16)$$

where we make explicit that the NDF and effective BRDF are functions of a texel q in the mipmap. The NDF considers all normals covered by that texel. Also note that this formulation works with arbitrary NDFs, and is not tied to mipmap-based filtering—we show an example with anisotropic filtering in Fig. 3.

Generality and Supported BRDFs: The form above is accurate for all BRDFs described by equation 5, including Lambertian, Blinn-Phong and measured microfacet distributions.[5] Moreover, it can be used even when the BRDF has an additional Fresnel or $g(\theta_d)$ factor, as per equation 6, since g need not be filtered.

5 Spherical Harmonics

We now discuss mathematical representations and algorithms that can be used for normal map filtering. The simplest approach is to work directly in spherical harmonics, using equation 16, as discussed in this section. Later, Sec. 7 discusses spherical vMFs.

5.1 Algorithm

At the finest level, the distribution function $\gamma(q)$ is a delta distribution at $\mathbf{n}(q)$, with[6] $\gamma_{lm}(q) = Y_{lm}(\mathbf{n}(q))$. An important insight is that,

[5] For some specular BRDFs, we also need to multiply by the cosine of the incident angle for a full transfer function. For the Spherical vMF method in Sec. 7, we simply multiply for each lobe by the cosine of the angle between light and lobe center (or effective normal). For the spherical harmonic method in Sec. 5, we simply use the mipmapped normals for the cosine term, since it is a relatively low-frequency effect.

[6] We use the real spherical harmonics. Otherwise, $\gamma_{lm}(q) = Y^*_{lm}(\mathbf{n}(q))$.

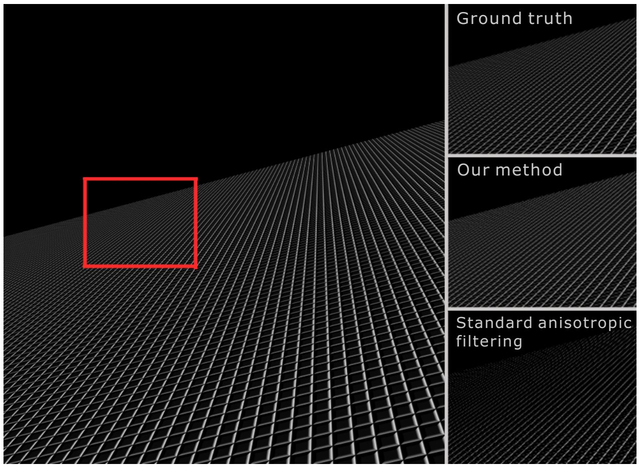


Figure 3: *Spherical harmonic anisotropic filtering for Lambertian reflection. Note the behavior for far regions of the plane. With standard normal filtering, these regions are averaged to a nearly flat surface. By contrast, our method is quite accurate in distant regions.*

unlike the original normals, these spherical harmonic NDF coefficients can now be *linearly* filtered or averaged. Hence, we can simply mipmap the spherical harmonic coefficients γ_{lm} in the standard way. Unlike previous approaches, *no non-linear fitting is required*.

At the time of rendering, we know the BRDF coefficients A_l. For many analytic models like Blinn-Phong, formulae for A_l are known [Ramamoorthi and Hanrahan 2001b]. For measured reflectance, A_l is obtained directly by a spherical harmonic transform of $\rho(\boldsymbol{\omega}\cdot\mathbf{n})$. The effective BRDF is then given by equation 16 as

$$\rho^{\mathrm{eff}}(\boldsymbol{\omega},q)=\sum_{l=0}^{l^*}\sum_{m=-l}^{l}A_l\gamma_{lm}(q)Y_{lm}(\boldsymbol{\omega}). \qquad (17)$$

For shading, assume a single point light source for now. At each surface location, we know the incident and outgoing directions, so it is easy to find the half-angle $\boldsymbol{\omega}_h$ or other parameterization $\boldsymbol{\omega}$, and then simply use the BRDF formula above directly for rendering.

We implement equation 17 in a pixel shader using GLSL. The spherical harmonics Y_{lm} are stored in floating point textures for table lookup, as are the mipmapped NDF coefficients $\gamma_{lm}(q)$. At runtime, we simply sum the coefficients, directly as per equation 17. Real-time frame rates are achieved comfortably for up to 64 spherical harmonic terms ($l^*\leq 7$, corresponding to a Blinn-Phong exponent $s\leq 12$ or a surface roughness $\sigma\geq 0.2$).

5.2 Results

Lambertian Reflection: In the Lambertian case, using only nine spherical harmonic coefficients ($l\leq 2$) suffices. An example is shown in Fig. 3, where (for this figure only) we also use GPU-based anisotropic filtering, instead of mipmapping, to show generality. Note the more accurate results for far away regions of the tile, where standard averaging of the normal produces an almost flat surface that is much darker than the actual (as illustrated in Fig. 1e).

Low-Frequency Specularities and Measured Reflectance: For specular materials with BRDF $\rho(\boldsymbol{\omega}_h\cdot\mathbf{n})$, the same approach can be applied. The BRDF can also be changed at run-time, since the NDF is independent of it. We have factored all of the materials in the database of [Matusik et al. 2003], using the $f(\theta_h)g(\theta_d)$ factorization in [Lawrence et al. 2006]. Figure 4 shows two examples of different materials, which we can switch between at runtime.

Figure 5 shows closeup views from an animation sequence of cloth draping over a sphere, using the blue fabric material from the Matusik database. Note the accuracy of our method (compare (b) with the ground truth in (c)). Also note the smooth transition between close (unfiltered) and distant (fully filtered) regions in (a) and (b), as well as the filtered zoomed out view in (d).

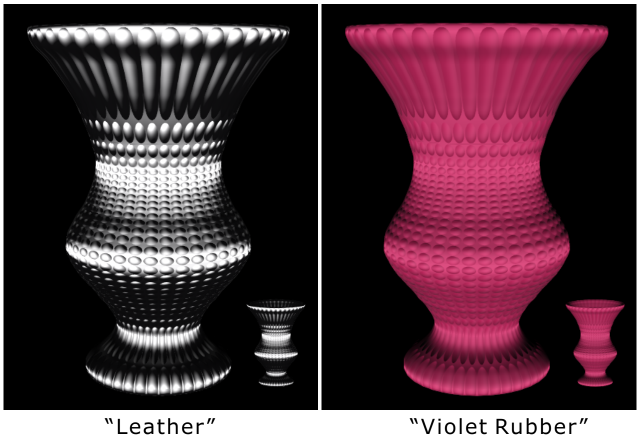


Figure 4: *The spherical harmonic method for normal mapping, with two of the materials in the Matusik database—we can support general measured BRDFs and change reflectance or material in real time. Notice also the correct filtering of the zoomed out view, shown at the bottom right.*

Discussion and Limitations: The spherical harmonic method is a simple practical approach for low-frequency materials. Unlike previous techniques, all operations are linear—no nonlinear fitting is required, and we can handle arbitrary lobe shapes (or functions $\rho(\boldsymbol{\omega}_h\cdot\mathbf{n})$). Moreover, the BRDF is decoupled from the NDF, enabling simultaneous changes of BRDF, lighting and viewpoint.

As with all low-frequency approaches, the spherical harmonic method requires many terms for high-frequency specularities (a Blinn-Phong exponent of $s=50$ needs about 300 coefficients). The following sections provide more practical solutions in these cases.

6 Spherically Symmetric Distributions

Spherical harmonics are general functions. To compactly represent high frequencies, we will instead use spherical distributions symmetric about a central direction (the average normal). These functions depend only on the radial distance from that center, and are not general 2D NDFs. However, by summing four to six such lobes, we can approximate general high-frequency NDFs (see Fig. 7). We now introduce a basic framework, followed in Sec. 7 by the algorithm with von Mises-Fisher (vMF) distributions [Fisher 1953].

6.1 Basic Theoretical Framework for using SRBFs

Consider a single basis function γ for the NDF, symmetric about some orientation or "normal" $\boldsymbol{\mu}$. For now, γ is a general *Spherical Radial Basis Function* (SRBF), of which vMFs are a special form (to be introduced later in Sec. 7). Equation 13 now becomes

$$\rho^{\mathrm{eff}}(\boldsymbol{\omega}\cdot\boldsymbol{\mu})=\int_{S^2}\gamma(\mathbf{n}\cdot\boldsymbol{\mu})\rho(\boldsymbol{\omega}\cdot\mathbf{n})\,d\mathbf{n}. \qquad (18)$$

It can be shown (for example, see [Tsai and Shih 2006]) that ρ^{eff} is itself radially symmetric about $\boldsymbol{\mu}$ (hence the form $\rho^{\mathrm{eff}}(\boldsymbol{\omega}\cdot\boldsymbol{\mu})$ above), and its spherical harmonic coefficients are given by

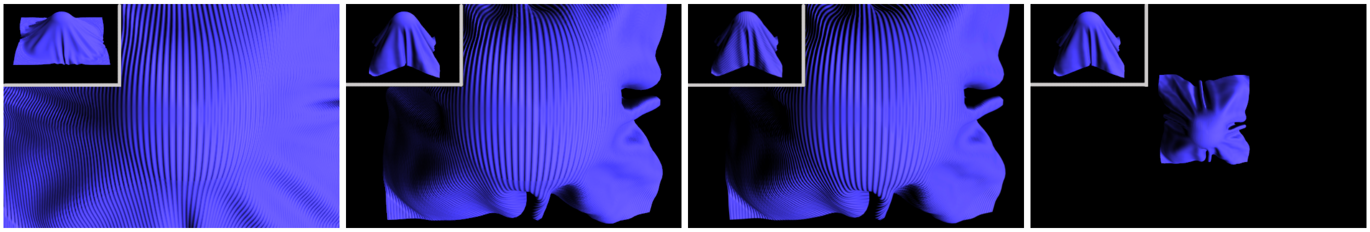
$$\boxed{\rho_l^{\mathrm{eff}}=A_l\gamma_l.} \qquad (19)$$

Compared to equation 16, this is essentially a simpler 1D convolution, since all functions are radially symmetric.

For rendering, we need to expand the effective BRDF in spherical harmonics, analogous to equation 17, but using only the Y_{l0} terms. Considering the summation of multiple lobes j, we obtain

$$\rho^{\mathrm{eff}}(\boldsymbol{\omega},q)=\sum_{j=1}^{J}\sum_{l=0}^{\infty}A_l\gamma_{l,j}(q)Y_{l0}(\boldsymbol{\omega}\cdot\boldsymbol{\mu}_j), \qquad (20)$$

where we again make clear that the NDF $\gamma_{l,j}$ is a function of the texel q. This equation can be used directly for shading, after we find $\boldsymbol{\omega}$ for the light source and view direction. Our actual implementation is given in Sec. 7, in terms of the vMF distributions.



(a) Our method, frame 1 (b) Our method, frame 2 (c) Ground truth, frame 2 (d) Our method, zoomed out

Figure 5: *Stills from a sequence of cloth draping over a sphere, with closeups indicating correct normal filtering (the full movie is shown in the video). Note the smooth transition from the center (almost no filtering) to the corners (fully filtered) in (b)—compare also with ground truth in (c). (d) is a zoomed out view that also filters correctly. We use a blue fabric material from the Matusik database as the BRDF.*

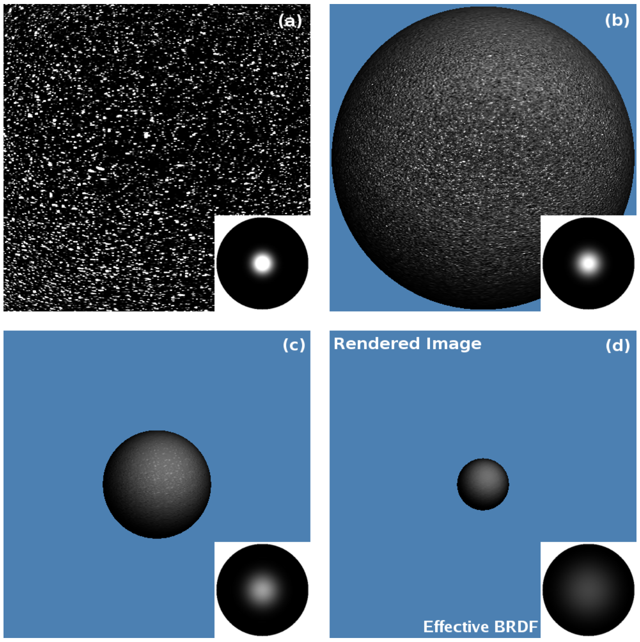


Figure 6: *Illustration of multiscale filtering of the BRDF/NDF. (a) shows a zoom into the sphere, where we see the individual facets and a sharp NDF/effective BRDF (sphere in inset). (b) zooms out to show the full sphere. The geometry now appears smoother, although roughness is still clearly visible. The effective BRDF, or convolution of NDF and BRDF, is now blurred, incorporating no-longer visible fine-scale geometry. As we zoom further out in (c) and (d), the geometry appears even smoother (with a smooth highlight), while the BRDF is further filtered (note wider lobes in (c) and (d)).*

6.2 Discussion: Unifying Framework and Multiscale

Our theoretical framework above unifies many normal filtering algorithms. Previous “peak or lobe-fitting” methods can be seen as special cases. For instance, [Schilling 1997; Toksvig 2004] effectively use a single lobe ($J = 1$), while [Fournier 1992] uses multiple Phong lobes for γ. These methods have generally adopted simple heuristics in terms of the BRDF. By developing a general convolution framework, we show how to separate the NDF from the BRDF, and properly account for general BRDFs A_l, which can even be changed on the fly—in contrast, even [Tan et al. 2005] is limited to Gaussian Torrance-Sparrow BRDFs that are predetermined.

Equation 19 also has an interesting multi-scale interpretation, as depicted in Fig. 6 for zooming out of a rough surface (using a single vMF lobe). At the finest scale (a), the geometry $\boldsymbol{\mu}$ is the original highest-resolution normal map $\mathbf{n}$. The NDF γ is a delta distribution at each texel, and the effective BRDF $\rho_l^{\text{eff}} = \rho_l$ (sharpest highlight on sphere in bottom right). At coarser scales, the geometry $\boldsymbol{\mu}$ is effectively a filtered or average version of the fine-scale normal map, with the geometry becoming smoother from (b)-(d). Note that (c) and (d) even have smooth highlights, without significant roughness. The effective BRDF is filtered by the NDF, essentially replacing the fine-scale geometry with a blurring of the BRDF.

Also note the symmetry between the BRDF and NDF in equation 19. While the common fine-scale interpretation is for a delta function NDF and the original BRDF, we can also view it as a delta function BRDF and an NDF given by A_l. These interpretations are consistent with most microfacet BRDF models, that start by assuming a mirror-like BRDF (delta function) and a complex microgeometry or NDF (microscopic V-grooves), and derive a net glossy BRDF on a smooth macro surface (delta function NDF).

7 Spherical vMF Algorithm

We now describe our algorithm to determine the central directions $\boldsymbol{\mu}_j$, and compute a sum of lobes to represent the NDF. A number of approaches have been proposed for PRT, but are not suitable here.

One method is to use zonal harmonics [Sloan et al. 2005]. However, our NDFs are not low-frequency, making fitting difficult, and storage inefficient since l is large. An alternative is to use Gaussian RBFs for γ [Tan et al. 2005], with parameters chosen using expectation maximization (EM) [Dempster et al. 1977; Bilmes 1997]. In this case, we simply need to store 3 parameters, for the amplitude, width and central direction. Whereas [Tan et al. 2005] pursued this approach using Euclidean (and therefore distorted) RBFs, we consider NDFs represented on their natural spherical domain, which also enables us to derive a simple convolution formula.

Indeed, *spherical* Gaussian RBFs, such as in [Tsai and Shih 2006] are most appropriate. However, nonlinear minimization for fitting is inefficient, given that we need to do so at each texel. Instead, we use a spherical variant [Banerjee et al. 2005] of EM, with the von Mises-Fisher[7] (vMF) distribution [Fisher 1953]. We also extend the basic spherical EM algorithm to handle color and different materials, create coherent lobes for hardware interpolation, and enable fast spherical harmonic convolution for rendering.

7.1 Estimation of Mixture of vMFs

First, we must estimate the mixture of vMF lobes for each texel using EM. The vMF probability distribution function is

$$\gamma(\mathbf{n}\cdot\boldsymbol{\mu};\theta) = \frac{\kappa}{4\pi\sinh(\kappa)}\exp[\kappa(\mathbf{n}\cdot\boldsymbol{\mu})], \qquad (21)$$

where the parameters $\theta = \{\kappa, \boldsymbol{\mu}\}$ are the inverse width κ and central direction $\boldsymbol{\mu}$. A *mixture of vMFs* (movMF) is defined as an affine combination of vMF lobes θ_j, with amplitude α_j, where $\sum_j \alpha_j = 1$,

$$\gamma(\mathbf{n};\Theta) = \sum_{j=1}^{J} \alpha_j \gamma_j(\mathbf{n}\cdot\boldsymbol{\mu}_j;\theta_j). \qquad (22)$$

We use spherical EM (algorithm 1) to fit a movMF to the normals covered at each texel in the mipmap. Line 5 of algorithm 1 shows the E-step. For all normals $\mathbf{n}_i$ in a given texel, we compute the expected likelihood $\langle z_{ij}\rangle$ that $\mathbf{n}_i$ corresponds to lobe j. Lines 9-14 execute the M-step, which computes maximum likelihood estimates of the parameters. In practice, we seldom need more than

[7] For the unit 3D sphere, this function is also known as the Fisher distribution, while for the unit 2D sphere (circle), it is known as the von Mises distribution. We use the more general term von Mises-Fisher distribution or vMF, that also applies to arbitrary dimension hyperspheres.

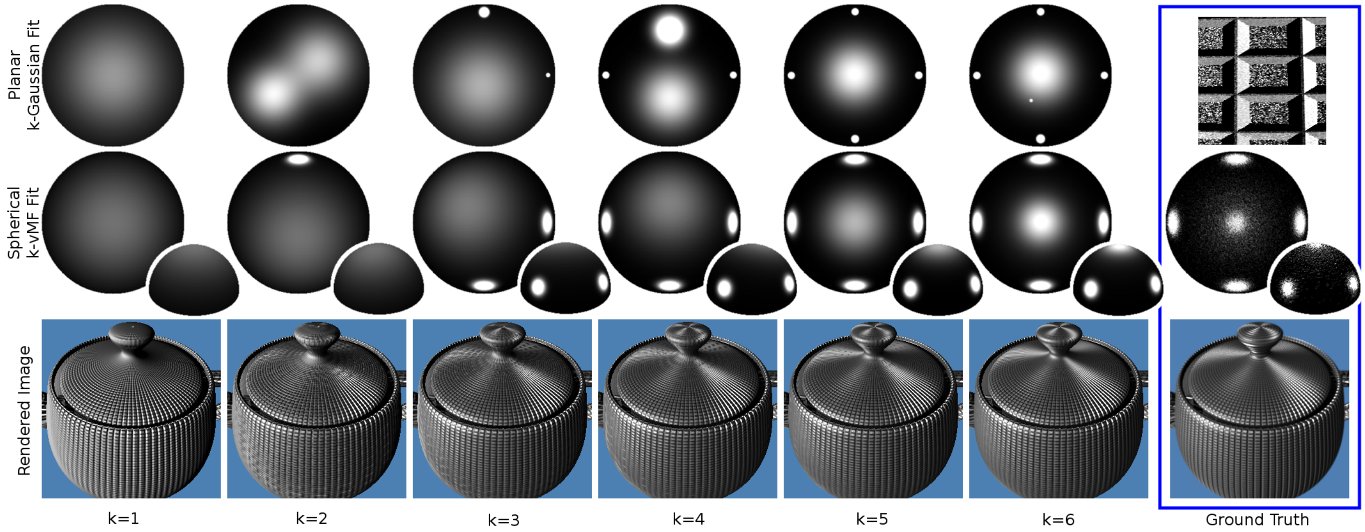


Figure 7: *Fitting of a spherical NDF from one of the texels in the mipmap with increasing numbers of vMF lobes (middle row). With 3-4 lobes, we already get excellent agreement in the rendered image. Note that each vMF lobe is symmetric about some central direction, and is fit on the natural spherical domain (which is why we show both a top and side view in the middle row). By contrast, a planar Gaussian fit (top row), as in Tan et al., must remain symmetric in the distorted planar space, and therefore has considerable errors at the boundaries of the hemisphere. Because no explicit convolution formula exists in the planar case, we only show renderings with our method (bottom row), which accurately match ground truth for a small number of vMF lobes.*

10 iterations, so the full EM algorithm for a 512×512 normal map converges in under 2 minutes. Note that this is an offline computation that needs to be done only once per normal map—unlike most previous work, it is also independent of the BRDF (and lighting).

Note the use of auxiliary variable $\mathbf{r}_j$ in line 11, which represents $\frac{\langle \mathbf{x}_j \rangle}{\alpha_j}$, where $\langle \mathbf{x}_j \rangle$ is the expected value of a random vector generated according to the scaled vMF distribution $\gamma(\mathbf{x};\theta_j)$. The central normal $\boldsymbol{\mu}_j$ and the concentration parameter κ_j are related to $\mathbf{r}_j$ by

$$\begin{aligned} \mathbf{r} &= A(\kappa)\boldsymbol{\mu}, \\ \text{where } A(\kappa) &= \coth(\kappa) - \frac{1}{\kappa}. \end{aligned} \qquad (23)$$

The direction $\boldsymbol{\mu}$ is found simply by normalizing $\mathbf{r}$ (line 13), while κ is given by $A^{-1}(\|\mathbf{r}\|)$; since no closed-form expression exists for A^{-1}, line 12 uses the approximation in [Banerjee et al. 2005].

Since EM is an iterative method, good initialization is important. For normal map filtering, we can proceed from the finest texels to coarser levels. At the finest level, we have only a single normal at each texel[8], so we need only a single lobe and directly set $\alpha = 1$, $\boldsymbol{\mu} = \mathbf{n}$, and κ to a large initial value. Assume we fit J lobes. At each coarser level of the mipmap, we will be combining normals from 4 finer level texels (a total of $4J$ lobes) to obtain J new lobes. A good initialization would be to choose the furthest J lobes of the finer level $4J$ vMFs. This is achieved using Hochbaum-Shmoys clustering [Hochbaum and Shmoys 1985; Agarwal et al. 2003] which is simple and nearly optimal. Note that the finer level lobes are used only for initialization—the actual fitting uses all normals covered by that texel in the mipmap.

The accuracy of our method is shown in Fig. 7, where we see that about four lobes suffices in most cases, with excellent agreement with six lobes. We also compare with the planar Gaussian EM fits of [Tan et al. 2005]. Because they work on a distorted planar projection of the hemisphere (top row), they have a significant loss of accuracy near the boundaries. Our method (middle row) works on the natural spherical domain (hence the side view shown), and is able to correctly create undistorted symmetric lobes anywhere on the sphere. Also note that [Tan et al. 2005] do not have an explicit convolution formula, while our method can be combined with any BRDF to produce accurate rendered images (bottom row).

Algorithm 1 The Spherical EM algorithm

```
1:  repeat
2:    {The E-step}
3:    for all samples $\mathbf{n}_i$ do
4:      for $j = 1$ to $J$ do
5:        $\langle z_{ij} \rangle \leftarrow \frac{\gamma_j(\mathbf{n}_i;\theta_j)}{\sum_{k=1}^{J} \gamma_k(\mathbf{n}_i;\theta_k)}$ {Expected likelihood of $\mathbf{n}_i$ in lobe $j$}
6:      end for
7:    end for
8:    {The M-step}
9:    for $j = 1$ to $J$ do
10:     $\alpha_j \leftarrow \frac{\sum_{i=1}^{N} \langle z_{ij} \rangle}{N}$
11:     $\mathbf{r}_j \leftarrow \frac{\sum_{i=1}^{N} \langle z_{ij} \rangle \mathbf{n}_i}{\sum_{i=1}^{N} \langle z_{ij} \rangle}$ {Auxiliary variable for $\kappa, \boldsymbol{\mu}$ in equation 23}
12:     $\kappa_j \leftarrow \frac{3\|\mathbf{r}_j\| - \|\mathbf{r}_j\|^3}{1 - \|\mathbf{r}_j\|^2}$
13:     $\boldsymbol{\mu}_j \leftarrow \mathrm{normalize}(\mathbf{r}_j)$
14:   end for
15: until convergence
```

7.2 Spherical Harmonic Coefficients for Rendering

For rendering, we will need the spherical harmonic coefficients γ_l of a normalized vMF lobe. These coefficients do not appear easy to find in the literature, so we derive them here based on reasonable approximations. First, for large κ, $\sinh(\kappa) \approx \exp[\kappa]/2$. In practice, this approximation is accurate as long as $\kappa > 2$, which is almost always the case. Hence, the vMF in equation 21 becomes

$$\gamma(\mathbf{n} \cdot \boldsymbol{\mu};\theta) \approx \frac{\kappa}{2\pi} \exp[-\kappa(1 - \mathbf{n} \cdot \boldsymbol{\mu})]. \qquad (24)$$

Let β be the angle between $\mathbf{n}$ and $\boldsymbol{\mu}$. Then, $1 - \mathbf{n} \cdot \boldsymbol{\mu} = 1 - \cos\beta$. For moderate κ, β must be small for the exponential to be nonzero. In these cases, $1 - \cos\beta \approx \beta^2/2$, and we get a Gaussian form,

$$\gamma(\mathbf{n} \cdot \boldsymbol{\mu};\theta) \approx \frac{\kappa}{2\pi} \exp[-\frac{\kappa}{2}\beta^2]. \qquad (25)$$

In [Ramamoorthi and Hanrahan 2001b], the spherical harmonic coefficients of a Torrance-Sparrow model of a similar form are com-

[8] Our framework also supports analytic functions, in which case a fine-scale texel would have a continuous normal distribution instead of a discrete normal. This distribution can be sampled and used directly for EM.

puted. For notational simplicity, let $\Lambda_l = \sqrt{4\pi/(2l+1)}$. Then,

$$\gamma = \frac{\exp[-\beta^2/(4\sigma^2)]}{4\pi\sigma^2} \;\Rightarrow\; \Lambda_l \gamma_l = \exp[-(\sigma l)^2]. \qquad (26)$$

Comparing with equation 25, we obtain

$$\begin{array}{rcl} \sigma^2 & = & \dfrac{1}{2\kappa} \\ \Lambda_l \gamma_l & = & \exp[-\sigma^2 l^2] = \exp[-\dfrac{l^2}{2\kappa}]. \end{array} \qquad (27)$$

Since each vMF lobe is treated independently, and the constants α_j and BRDF coefficients can be multiplied separately, we focus on convolving the normalized BRDF with a single normalized vMF lobe. It is possible to directly use equation 27 for the vMF coefficients and equation 20 for rendering with general BRDFs.

However, a much simpler method is available for the important special forms of Blinn-Phong and Torrance-Sparrow like BRDFs. First, consider a normalized Blinn-Phong[9] model of the form,

$$\rho(\boldsymbol{\omega}_h \cdot \mathbf{n}) = \frac{s+1}{2\pi}(\boldsymbol{\omega}_h \cdot \mathbf{n})^s, \qquad (28)$$

where s is the specular exponent or shininess. It can be shown [Ramamoorthi and Hanrahan 2001b] that the spherical harmonic coefficients are $A_l \approx \exp[-l^2/2s]$. Therefore, the result after convolution with the vMF is still approximately a Blinn-Phong shape, with

$$\begin{array}{rcl} \dfrac{1}{2s'} = \dfrac{1}{2s} + \dfrac{1}{2\kappa} & \Longrightarrow & s' = \dfrac{\kappa s}{\kappa + s} \\ \rho^{\text{eff}}(\boldsymbol{\omega}_h \cdot \boldsymbol{\mu}) & = & \dfrac{s'+1}{2\pi}(\boldsymbol{\omega}_h \cdot \boldsymbol{\mu})^{s'}. \end{array} \qquad (29)$$

For a Torrance-Sparrow like BRDF of the form of equation 26, we obtain a similar form for ρ^{eff}, only with

$$\sigma' = \sqrt{\sigma^2 + (2\kappa)^{-1}}. \qquad (30)$$

The simplicity of these results mean it is trivial to change BRDF parameters on the fly, and to consider very high-frequency BRDFs.

7.3 Extensions

Different Materials/Colors: It is often the case that one would like to associate additional spatially varying properties (such as color, material blending weights, etc.) to a normal map. For example, the normal map in Fig. 2 contains regions of different colors. We represent these properties in a feature vector $\mathbf{y}_i$ associated with each normal $\mathbf{n}_i$, and also extend the EM algorithm accordingly.

For each vMF lobe, we would now like to find a $\mathbf{y}_j$ that best describes the $\mathbf{y}_i$ of all its underlying texels. In the appendix, we augment the EM likelihood function with an additional term, that can be maximized to yield an extra line in the M-step,

$$\mathbf{y}_j \leftarrow \frac{\sum_{i=1}^{N} \langle z_{ij} \rangle \mathbf{y}_i}{\sum_{i=1}^{N} \langle z_{ij} \rangle} \qquad (31)$$

Note that since $\mathbf{y}_j$ does not affect the E-step, the preceding can simply be run as a postprocess to the vanilla EM algorithm.

Coherent Lobes for Hardware Interpolation: One approach to trilinear mipmap filtering is simply to consider all 8 relevant texels, averaging the shading for each. Greater efficiency (usually a 2× to 4× speedup) is obtained if we first use hardware mipmapping to interpolate the parameters of each vMF lobe from all K neighbors. We can then simply run our GPU pixel shader once on the interpolated parameters. This requires that corresponding lobes of adjacent texels be similarly aligned, and that a new parameterization be chosen for accurate linear interpolation.

For alignment, we introduce a new term in our EM likelihood function, and maximize (the details are in the appendix). The final result replaces line 13 in the M-step of algorithm 1 with

$$\boldsymbol{\mu}_j \leftarrow \text{normalize}\left(\mathbf{r}_j + C \sum_{k=1}^{K} \alpha_{jk} \boldsymbol{\mu}_{jk}\right). \qquad (32)$$

C is a parameter that modulates the strength of the alignment term (which seeks to move $\boldsymbol{\mu}_j$ closer to the central directions $\boldsymbol{\mu}_{jk}$ of the K neighbors, preferring neighbors with large amplitudes α_{jk}.).

As in the planar Gaussian EM method of [Tan et al. 2005], we build our aligned movMFs starting at the topmost (that is, most filtered) mipmap level and proceed downward, following scanline ordering within each individual level. In the interest of performance, we use only previously computed texels as neighbors.

We next consider trilinear interpolation of the variables. Unfortunately, the customary vMF parameters $\{\kappa, \boldsymbol{\mu}\}$ control non-linear aspects of the vMF lobe, and do not interpolate linearly. To solve this problem, we recall from Sec. 7.1 that $\boldsymbol{\mu}$ and κ can be inferred from the scaled Euclidean mean $\mathbf{r} = \frac{\langle \mathbf{x} \rangle}{\alpha}$ of a given vMF distribution. By linearity of expectation, we can interpolate $\alpha \mathbf{r} = \langle \mathbf{x} \rangle$ linearly, as well as the amplitude α,

$$\begin{array}{rcl} \tilde{\alpha}_j & = & T(\alpha) \\ \tilde{\mathbf{r}}_j & = & T(\alpha_j \mathbf{r}_j)/T(\alpha), \end{array} \qquad (33)$$

where $T(\cdot)$ denotes hardware interpolation (or trilinear filtering), and $\{\tilde{\kappa}_j, \tilde{\boldsymbol{\mu}}_j\}$ can be easily found in a fragment shader using lines 12 and 13 of algorithm 1. For implementation, we store the jth lobe θ_j of each movMF in a standard texture mipmap using one channel for α_j and one channel each for the three components of $\alpha_j \mathbf{r}_j$. Color/material properties $\mathbf{y}_j$ are stored in corresponding textures.

7.4 Results

Figure 2 shows the accuracy of our method, and makes comparisons to ground truth and alternative techniques. It also shows our ability to use different materials for different parts of the normal map.

Our formulation allows for general and even dynamically changing BRDFs. Figure 8 shows a complex scene, where the reflectance changes over time, decreasing in shininess (intended to simulate drying using the model in [Gu et al. 2006]). Although not shown, the lighting and view can also vary—the bottom row shows close-ups with different illumination. Note the correct filtering for dinosaurs in the background, and for further regions along the neck and body of the foreground dinosaur. Even where individual bumps are not visible, the overall change in appearance as the reflectance changes is clear. This complex scene has 14,898 triangles for the dinosaurs, 139,392 triangles for the terrain and 6 textures for the normal maps and dinosaur skins. It renders at 75 frames per second at a resolution of 800x600 on an NVIDIA 8800 graphics card. In this example, we used six unaligned vMF lobes, with both diffuse and specular shading implemented as a simple fragment shader.

8 Complex Lighting

Our vMF-based normal map filtering technique can also be extended to complex environment map lighting.[10] Equation 4 is a convolution that becomes a simple dot product in spherical harmonics,

$$B(\boldsymbol{\mu}) = \int_{S^2} L(\boldsymbol{\omega}_i)\rho^{\text{eff}}(\boldsymbol{\omega} \cdot \boldsymbol{\mu})\, d\boldsymbol{\omega}_i, \qquad (34)$$

where the effective BRDF ρ^{eff} is the convolution of the vMF lobe with the BRDF, and $\boldsymbol{\mu}$ is the central direction of the vMF lobe (effective "normal") as usual. For the diffuse component $\boldsymbol{\omega} = \boldsymbol{\omega}_i$,

[9] Note that the diffuse component can also be handled in a very similar way, simply setting $s = 1$ and $\boldsymbol{\omega} = \boldsymbol{\omega}_i$.

[10] The direct spherical harmonic method in Sec. 5 is more difficult to apply, since general spherical harmonics cannot be rotated as easily as radially symmetric functions between local and global frames.

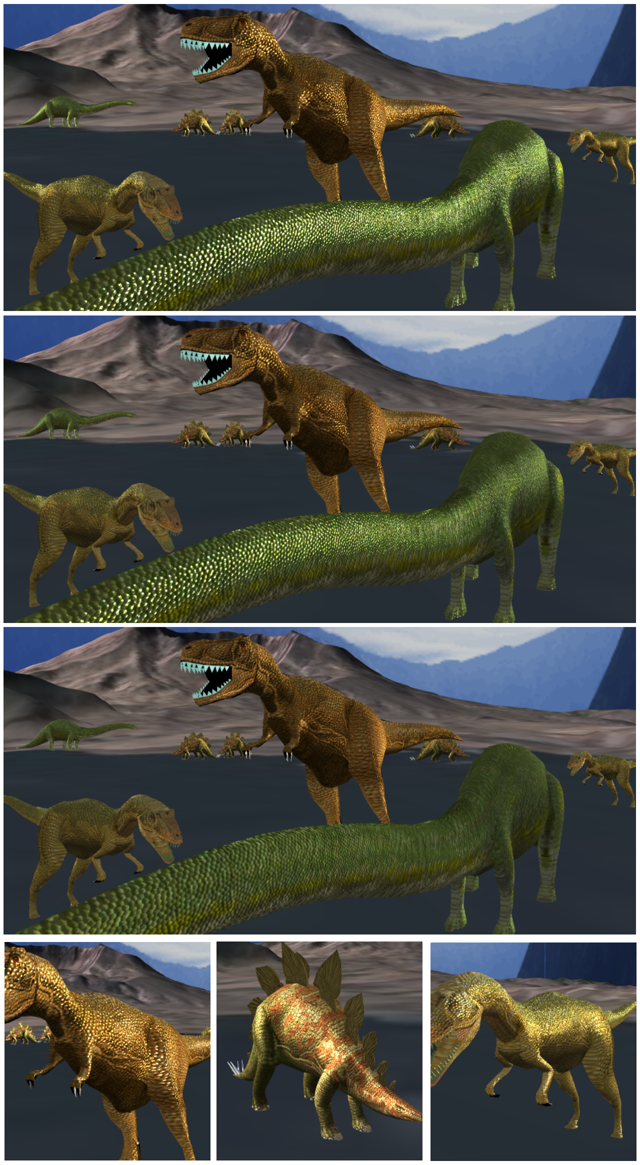


Figure 8: Our framework can handle complex scenes, allowing for general reflectance, which can even be changed at run-time. Here, the BRDF becomes less shiny over time. Note the correct filtering and overall changes in appearance for further regions of the foreground dinosaur, and those in background. The bottom row shows closeups (when the material is shiny) with a different lighting condition. This example also shows that we can combine normal maps with standard texture mapping.

and the spherical harmonic coefficients can simply be multiplied according the convolution formula, $B_{lm} = \Lambda_l \rho_l^{\text{eff}} L_{lm}$, so that

$$B = \sum_{l=0}^{l^*} \sum_{m=-l}^{l} \Lambda_l \rho_l^{\text{eff}} L_{lm} Y_{lm}(\boldsymbol{\mu}). \qquad (35)$$

For the specular component however, the BRDF is expressed in terms of $\boldsymbol{\omega} = \boldsymbol{\omega}_h$, and we need to change the variable of integration in equation 34 to $\boldsymbol{\omega}_h$ (which leads to a factor $4(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_h)$),

$$\begin{aligned} B(\boldsymbol{\mu}) &= \int_{S^2} \left[L(\boldsymbol{\omega}_i(\boldsymbol{\omega}_h, \boldsymbol{\omega}_o)) \cdot 4(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_h)\right] \rho^{\text{eff}}(\boldsymbol{\omega}_h \cdot \boldsymbol{\mu})\, d\boldsymbol{\omega}_h \\ &= \int_{S^2} L'(\boldsymbol{\omega}_h) \rho^{\text{eff}}(\boldsymbol{\omega}_h \cdot \boldsymbol{\mu})\, d\boldsymbol{\omega}_h. \qquad (36) \end{aligned}$$

Thus, we simply need to consider a new reparameterized lighting $L'(\boldsymbol{\omega}_h) = L(\boldsymbol{\omega}_i(\boldsymbol{\omega}_h, \boldsymbol{\omega}_o)) \cdot 4(\boldsymbol{\omega}_i \cdot \boldsymbol{\omega}_h)$. As the half angle depends on both viewing and lighting angles ($\boldsymbol{\omega}_o$ and $\boldsymbol{\omega}_i$), the above integration implicitly limits us to a fixed view with respect to the lighting. To interactively rotate the lighting, we precompute a sparse set of rotated lighting coefficients and interpolate the shading.

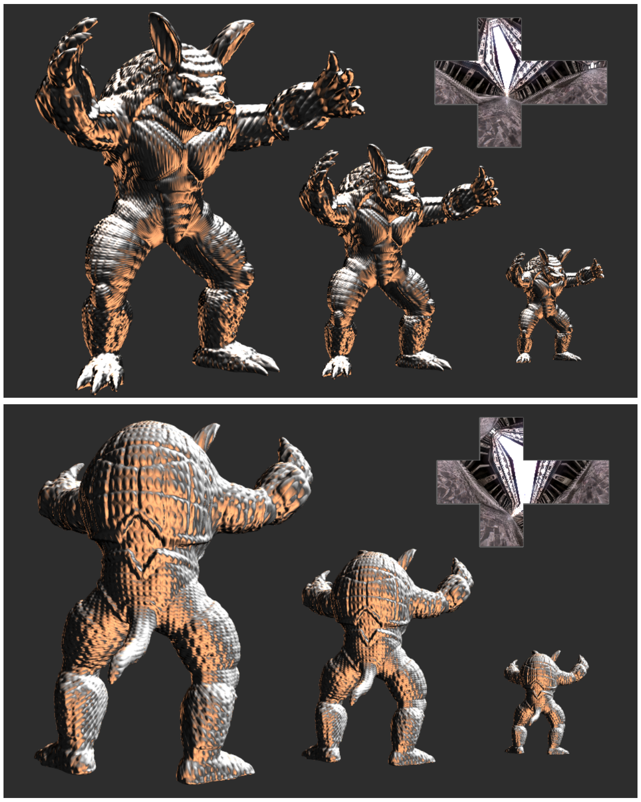


Figure 9: Armadillo model with 350,000 polygons rendered in real time with normal maps in dynamic environment lighting. We use 6 vMF lobes, and spherical harmonics up to order 8 for the specular component.

Finally, in analogy with equation 35,

$$B = \sum_{l=0}^{l^*} \sum_{m=-l}^{l} \Lambda_l \rho_l^{\text{eff}} L'_{lm} Y_{lm}(\boldsymbol{\mu}). \qquad (37)$$

Figure 9 shows an image of an armadillo, with approximately 350,000 polygons and a normal map, rendered at real-time rates in dynamic environment lighting. We are able to render interactively with up to 6 vMF lobes and $l^* = 8$ in equation 37.

9 Conclusions and Future Work

We have developed a comprehensive theoretical framework for normal map filtering with many common types of reflectance models. Our method is based on a new analytic formulation of normal map filtering as a convolution of the NDF and BRDF. This leads to practical algorithms using spherical harmonics and spherical vMFs, that enable general reflectance functions. The methods are implemented in real-time rates in graphics hardware as simple GPU shaders.

We believe this paper also makes broader contributions to many areas of rendering, and beyond. The convolution result unifies a geometric problem (normal mapping) with understanding of lighting and BRDF interaction in appearance. The connection between spherical harmonics and spherical vMFs may also allow a better understanding of, and new practical algorithms even for PRT.

Moreover, we introduce spherical EM and vMF distributions into computer graphics, where they will likely find many other applications. For instance, vMF lobes are radially symmetric functions that are well suited for BRDF importance sampling [Lawrence et al. 2004] for global illumination of normal mapped objects—our experiments indicate they produce sharper and higher-quality results than standard mipmap filtering even for offline rendering.

In summary, normal mapping is an old technique, but correct filtering has been a challenging problem, because shading is nonlinear in the surface normal. In this paper, we have show how advanced mathematical analysis can shed important new insight, and take a significant step towards addressing this long-standing problem.

References

Agarwal, S., Ramamoorthi, R., Belongie, S., and Jensen, H. 2003. Structured importance sampling of environment maps. *ACM Transactions on Graphics (SIGGRAPH 2003) 22*, 3, 605–612.

Banerjee, A., Dhillon, I., Ghosh, J., and Sra, S. 2005. Clustering on the unit hypersphere using von Mises-Fisher distributions. *Journal of Machine Learning Research 6*, 1345–1382.

Basri, R., and Jacobs, D. 2001. Lambertian reflectance and linear subspaces. In *International Conference on Computer Vision*, 383–390.

Becker, B., and Max, N. 1993. Smooth transitions between bump rendering algorithms. In *SIGGRAPH 93*, 183–190.

Bilmes, J. 1997. A gentle tutorial on the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Tech. Rep. ICSI-TR-97-021, UC Berkeley ICSI.

Blinn, J. 1978. Simulation of wrinkled surfaces. In *SIGGRAPH 78*, 286–292.

Cohen, J., Olano, M., and Manocha, D. 1998. Appearance preserving simplification. In *SIGGRAPH 98*, 115–122.

Crow, F. 1984. Summed-area tables for texture mapping. In *SIGGRAPH 84*, 207–212.

Dempster, A., Laird, N., and Rubin, D. 1977. Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1–38.

Donnelly, W. 2005. Per-pixel displacement mapping with distance functions. In *GPU Gems 2 (Chapter 8)*, 123–136.

Fisher, R. 1953. Dispersion on a sphere. *Proceedings of the Royal Society of London, Series A 217*, 295–305.

Fournier, A. 1992. Normal distribution functions and multiple surfaces. In *Graphics Interface Workshop on Local Illumination*, 45–52.

Gu, J., Tu, C., Ramamoorthi, R., Belhumeur, P., Matusik, W., and Nayar, S. 2006. Time-varying surface appearance: Acquisition, modeling and rendering. *ACM Transactions on Graphics (SIGGRAPH 2006) 25*, 3, 762–771.

Hochbaum, D., and Shmoys, D. 1985. A best possible heuristic for the k-center problem. *Mathematics of Operations Research.*

Horn, B. K. P. 1984. Extended gaussian images. *Proceedings of the IEEE 72*, 1671–1686.

Lalonde, P., and Fournier, A. 1997. A wavelet representation of reflectance functions. *IEEE TVCG 3*, 4, 329–336.

Lawrence, J., Rusinkiewicz, S., and Ramamoorthi, R. 2004. Efficient BRDF importance sampling using a factored representation. *ACM Transactions on Graphics (SIGGRAPH 2004) 23*, 3.

Lawrence, J., BenArtzi, A., Decoro, C., Matusik, W., Pfister, H., Ramamoorthi, R., and Rusinkiewicz, S. 2006. Inverse shade trees for non-parametric material representation and editing. *ACM Transactions on Graphics (SIGGRAPH 2006) 25*, 3, 735–745.

Matusik, W., Pfister, H., Brand, M., and McMillan, L. 2003. A data-driven reflectance model. *ACM Transactions on Graphics (SIGGRAPH 03 proceedings) 22*, 3, 759–769.

Ng, R., Ramamoorthi, R., and Hanrahan, P. 2004. Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics (SIGGRAPH 2004) 23*, 3, 475–485.

Olano, M., and North, M. 1997. Normal distribution mapping. Tech. Rep. 97-041, UNC.

Ramamoorthi, R., and Hanrahan, P. 2001. An efficient representation for irradiance environment maps. In *SIGGRAPH 01*, 497–500.

Ramamoorthi, R., and Hanrahan, P. 2001. A signal-processing framework for inverse rendering. In *SIGGRAPH 01*, 117–128.

Schilling, A. 1997. Toward real-time photorealistic rendering: Challenges and solutions. In *SIGGRAPH/Eurographics Workshop on Graphics Hardware*, 7–16.

Sloan, P., Kautz, J., and Snyder, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM TOG (SIGGRAPH 02 proceedings) 21*, 3, 527–536.

Sloan, P., Luna, B., and Snyder, J. 2005. Local, deformable precomputed radiance transfer. *ACM Transactions on Graphics (SIGGRAPH 05 proceedings) 24*, 3, 1216–1224.

Sloan, P. 2006. Normal mapping for precomputed radiance transfer. *Symposium on Interactive 3D Graphics*, 23–26.

Strehl, A., Ghosh, J., and Mooney, R. 2000. Impact of similarity measures on web-page clustering. In *Proc Natl Conf on Artificial Intelligence : Workshop of AI for Web Search (AAAI 2000)*, 58–64.

Tan, P., Lin, S., Quan, L., Guo, B., and Shum, H. 2005. Multiresolution reflectance filtering. In *EuroGraphics Symposium on Rendering 2005*, 111–116.

Toksvig, M. 2004. Mipmapping normal maps. Tech. Rep. TB-01-256-0001, nVIDIA.

Tsai, Y., and Shih, Z. 2006. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transactions on Graphics (SIGGRAPH 2006) 25*, 3, 967–976.

Wang, L., Wang, X., Tong, X., Lin, S., Hu, S., Guo, B., and Shum, H. 2003. View-dependent displacement mapping. *ACM Transactions on Graphics (SIGGRAPH 2003) 22*, 3, 334–339.

Williams, L. 1983. Pyramidal parametrics. In *SIGGRAPH 83*, 1–11.

Appendix: Spherical EM Extensions

In this appendix, we briefly describe the likelihood function for spherical EM, and how we augment it for colors/materials and coherent lobes. The net likelihood function is a product of 3 terms,

$$P(X,Z|\Theta)P(Y,Z|\Theta))P(\Theta|N(\Theta)),$$

where X are the samples (in this case input normals), Z are the hidden variables (in this case which vMF lobe a sample X is drawn from), Θ are parameters for all vMF lobes and $N(\Theta)$ are parameters for neighbors. The first factor corresponds to standard spherical EM, the second factor corresponds to the colors/materials Y,

$$P(Y,Z|\Theta) = \prod_{i=1}^{N} \exp[-\|\mathbf{y}_{z_i} - \mathbf{y}_i\|^2], \tag{38}$$

and the final factor to coherent lobes for interpolation,

$$P(N(\Theta)|\Theta) = \prod_{j=1}^{J}\prod_{k=1}^{K} \exp[C'\alpha_{jk}(\boldsymbol{\mu}_j \cdot \boldsymbol{\mu}_{jk})]. \tag{39}$$

In EM, we seek to maximize the log likelihood,

$$\ln\left[P(X,Z|\Theta)P(Y,Z|\Theta))P(\Theta|N(\Theta))\right] = \sum_{i=1}^{N} \ln\gamma(\mathbf{n}_i|\theta_{z_i}) + \sum_{i=1}^{N} -\|\mathbf{y}_i - \mathbf{y}_{z_i}\|^2 + \sum_{j=1}^{J}\sum_{k=1}^{K} C'\alpha_{jk}(\boldsymbol{\mu}_j \cdot \boldsymbol{\mu}_{jk}). \tag{40}$$

Considering all J lobes and hidden variables $\langle z_{ij}\rangle$, we seek to maximize

$$\sum_{j=1}^{J}\left[\sum_{i=1}^{N} \ln\gamma(\mathbf{n}_i|\theta_j)\langle z_{ij}\rangle + \sum_{i=1}^{N} -\|\mathbf{y}_i - \mathbf{y}_{z_i}\|^2\langle z_{ij}\rangle + \sum_{k=1}^{K} C'\alpha_{jk}(\boldsymbol{\mu}_j \cdot \boldsymbol{\mu}_{jk})\right]$$

with respect to all parameters. Maximizing with respect to $\mathbf{y}_j$, we directly obtain equation 31. The maximization with respect to $\boldsymbol{\mu}_j$ is more complex,

$$\boldsymbol{\mu}_j = \text{normalize}\left(\kappa_j \sum_{i=1}^{N} \mathbf{n}_i\langle z_{ij}\rangle + C' \sum_{k=1}^{K} \alpha_{jk}\boldsymbol{\mu}_{jk}\right). \tag{41}$$

Finally, redefining $C = C'/\kappa_j$, we obtain equation 32.