A Smooth, Efficient Representation of Reflectance

Masaki Kameya
Washington State University
2710 University Drive
Richland, WA 99352
mkameya@tricity.wsu.edu

Robert R. Lewis
Washington State University
2710 University Drive
Richland, WA 99352
bobl@tricity.wsu.edu

March 24, 2002*

Abstract

We present a representation for reflectance, especially measured reflectance data, that is both smooth and time-efficient. By fitting a multi-level B-spline approximation to bidirectional reflectance distribution function (BRDF) data, we can obtain a continuous function that matches the data to arbitrary precision. This algorithm shows very good time efficiency both constructing as well as evaluating the fit. Synthesized images show the accuracy and the smoothness of the fit BRDF. The finer level of fitting we use, the higher accuracy we can achieve. Our results compare favorably with other schemes currently in wide use.

One shortcoming of our method is the amount of storage required. We present one approach that addresses this, a bi-level representation combined with hashing.

1 Introduction

In computer graphics, the interaction of light with a surface is typically described by a bidirectional reflectance distribution function (BRDF). The BRDF represents all visible properties of the surface at the sub-textural level.

A BRDF is a function of an incident direction $\mathbf{S}$ and a reflected direction $\mathbf{V}$, defined on the upper hemisphere $\Omega_{\mathbf{N}}$ surrounding the surface normal $\mathbf{N}$ symmetrically. It is a ratio of the reflected radiance dL_r in a given reflected direction to the irradiance dE coming from an incident direction. We denote the BRDF as $f_r(\mathbf{S}, \mathbf{V})$. Hence

$$dL_r = f_r(\mathbf{S}, \mathbf{V}) dE. \tag{1}$$

The irradiance dE of a source in direction $\mathbf{S}$ subtending a solid angle $d\omega_i$ can be represented as:

$$dE(\mathbf{S}) = L_i(\mathbf{S}) \cos\theta_i \, d\omega_i, \tag{2}$$

where $L_i(\mathbf{S})$ is the incident radiance in the direction $\mathbf{S}$, θ_i is the incident polar angle ($\cos\theta_i = \mathbf{N} \cdot \mathbf{S}$) and $d\omega_i$ is the amount of solid angle subtended by the source. The total reflected radiance L_r in the direction $\mathbf{V}$ is obtained by integrating the irradiance $dE(\mathbf{S})$ over all directions on $\Omega_{\mathbf{N}}$:

$$L_r(\mathbf{V}) = \int_{\Omega_{\mathbf{N}}} f_r(\mathbf{S}, \mathbf{V}) L_i(\mathbf{S}) \cos\theta_i \, d\omega_i. \tag{3}$$

Typically, the directions $\mathbf{S}$ and $\mathbf{V}$ are represented with spherical coordinates, i.e., a polar angle θ and an azimuthal angle ϕ. The BRDF f_r would then be a function of four parameters, $f_r(\theta_i, \phi_i, \theta_o, \phi_o)$, where (θ_i, ϕ_i) represents S and (θ_o, ϕ_o) represents V.

The BRDF for a perfectly diffuse reflector (a Lambertian surface) is constant, while an ideal mirror specular reflector has Dirac delta functions in its BRDF, with the only non-zero direction being the reflection of the viewing direction.

*This is an approximate reconstruction of a paper (no longer obtainable) presented at the Thirteenth Western Computer Graphics Symposium (Silver Star, British Columbia; March 24-27, 2002), which should be the cited reference.

There are two overall ways to derive a BRDF. One way yields an analytical formula, either from from an ad-hoc, intuitive model or a physically-based model. The other way is to collect data: measure the reflectance for a given material under laboratory conditions.

We present here a method to represent any (especially measured) BRDF data. We apply a modified form of the B-spline based approximation algorithm presented by Lee et al. [17] to fit the measured data.

In Section 2, we will review previous work on BRDF representation and explain the approximation algorithm we use. In Section 3, we describe the application of the approximation algorithm to publicly-available measured BRDF data and show the results of the fit. We improve the method by deriving a sparse representation which incorporates minimal perfect hashing to reduce storage and minimize time requirements in Section 4. Section 5 concludes the paper.

2 Previous Work

Many researchers have addressed the problem of representing reflectance. In this section, we will review some of the more popular representations and also briefly describe the approximation algorithm we will adapt for our use.

2.1 A Brief History of Reflectance Modeling

BRDFs used in computer graphics are produced in one of two ways: analytic derivation or direct measurement. We can further categorize the analytical derivation approach into two subclasses: ad-hoc models and physically-based models. We will discuss each category in roughly chronological order.

2.1.1 Ad-Hoc Analytical Models

Ad-hoc models are created out of a need to exhibit observed or expected behavior. Their derivation does not involve physical considerations.

Phong[23] presented a model that was designed to capture the behavior of roughened surfaces by raising a cosine to a user-specified power and scaling the result by a user-specified coefficient. This model is relatively old and not physically correct, but still widely used in computer graphics because it produces fairly good images, is computationally inexpensive, and, with some experience, is easy for the user to control.

Blinn[1] modified Phong's work by introducing a "halfway vector": the normalized mean of the viewing direction and incident light direction. Whether to use a Phong or a Phong-Blinn model is often the choice of the hardware or rendering package implementer.

Lewis[18] derived physically plausible representations of these models. "Physically plausible" is a weaker constraint than "physically based" in that it accepts forms of reflectance models which cannot be ruled out on the basis of energy conservation and reciprocity.

2.1.2 Physically-Based Analytical Models

As graphics capabilities expanded, so did the need for realism in images. We would, however, assert that physically-based models predate ad-hoc ones, as we must in fairness consider the work of Lambert[15] to be physically-based. Lambert's derivation was based upon reasoning about the nature of reflectance. Lambert's Law, which is that the amount of light reflected on the surface is independent of the viewer's direction and depends only on the angle between the incident light direction and the surface normal, is still widely used for diffuse objects and has even been "extended" by Oren and Nayar[22] to non-diffuse surfaces.

For non-diffuse cases, Cook and Torrance[3] derived a model based on geometrical optics, assuming specular V-grooves and incorporating masking and self-shadowing effects. Schlick[24] extended this model with a more comprehensive one that has seen considerable use in the literature.

He et al. [10] extended the Cook-Torrance model to the region of physical optics. This takes into account polarization, surface roughness, masking, and shadowing and is given by a single formula which consists of specular, directional diffuse, and uniform diffuse terms.

A physically-based model usually incurs greater computational cost than a non-physically-based one. Furthermore, it can be difficult to control – to "model" – physically-based parameters to obtain a desired material appearance.

2.1.3 Data-Driven Models

With the advent of image-based rendering in recent years, another popular way to represent reflectance has arisen: approximating or interpolating measured data or data computed from a more complex analytical model. The mathematical representation used derives neither from qualitative desirability or control needs as in the case of the ad-hoc methods nor from a rigorous physical derivation as in the case of the physically-based methods. Instead, it comes from finding an efficient and compact fit to the data itself: It is "data-driven". Some of these enforce physical plausibility, which is certainly desirable when fitting real-world data.

Ward[16] simplified the geometric attenuation and Fresnel factors from the Cook-Torrance model into one normalized parameter and incorporated them into a Gaussian lobe model. In addition, he incorporated two roughness parameters into an elliptical Gaussian model, successfully fitting data he had measured on his own apparatus.

Other popular representations use basis functions. As a BRDF is generally represented as a function of spherical coordinates, it is natural to use a basis function defined over spherical coordinates. Westin et al.[27] and Sillon[26] represented a BRDF as a weighted sum of spherical harmonics. Westin's model can represent both isotropic and anisotropic BRDFs. To obtain an accurate representation of specular data, however, these models require many terms.

Schröder and Sweldens[25] show how to build "second-generation wavelets" which make it possible to define wavelets on nearly arbitrary domains. As an example, they represent a BRDF with wavelets defined on sphere. Their implementation uses fewer coefficients than spherical harmonics and coefficients can be computed efficiently, but their approach only uses wavelets for two degrees of freedom. In addition, the evaluation time of their BRDF model depends on the wavelet's resolution.

Koenderink et al.[13] constructed an orthonormal basis based on the Cartesian product of the hemisphere by mapping Zernike polynomials onto a unit disk. They then project the BRDF into this vector space. Zernike polynomials appear to offer some advantages over spherical harmonics, but still require more terms to capture greater specularity.

Lafortune et al.[14] represented a BRDF as a non-linear summation of powers of cosine lobes. It may be considered a generalization of Phong's original formula. This model can represent important BRDF behavior such as off-specular reflection and a retro-reflection with small number of parameters and it is currently a very popular reflectance model. Owing to it's use of non-orthogonal fitting functions via a Levenberg-Marquardt approach,

Some researchers represent a BRDF as a product, rather than a summation, of functions.

Fournier[9] separates a BRDF into two functions, one a function of the incident direction and the other of the reflected direction, using the technique of singular value decomposition. He then sums the products of these two functions with weights and applied the results to a Phong model and to Ward's experimental data[16].

McCool et al.[21] also used this technique, decomposing a BRDF into products of two or more functions of lower dimensionality. For their implementation, they decompose an isotropic BRDF into three factors, with two different functions. The value of each function is stored as two-dimensional texture map and makes use of texturing hardware.

2.2 N-dimensional Multilevel B-Spline Approximation

There are many ways to fit data. We have chosen the multilevel B-spline algorithm of Lee et al. [17] (hereafter, "LWS") for these reasons:

- It does not require uniform sampling.
- It is fast to compute.
- It is fast to evaluate.
- Since it uses cubic B-splines, the fit is C^2 continuous.

LWS first develop a "basic approximation" (BA) algorithm. We describe here its essential details.

Let $\Psi = \{x_d | 0 \leq d < D, 0 \leq x_d < N\}$ be a hypercubic rectangular domain in $\mathcal{R}^D$. Consider a given set of scattered points $\{\mathbf{P}\}$ in Ψ. For this point set, we will find

the fit function f. f approximates data as a uniform, non-parametric[1], bi-cubic B-spline function, which is defined by a uniform control lattice Φ overlaid on the integer coordinates of Ψ (and extending slightly outside – see LWS for details).

Let $\phi_{\mathbf{j}}$ be the coefficient (which may also be thought of as a "control point") at $\mathbf{j}$ on Φ. f is then

$$f(x_0, \ldots, x_{D-1}) = \sum_{k_0=0}^{3} \cdots \sum_{k_{D-1}=0}^{3} \phi_{(k_0+i_0)\cdots(k_{D-1}+i_{D-1})} B_{k_0}(s_0) \cdots B_{k_{D-1}}(s_{D-1}), \quad (4)$$

where $i_\alpha = \lfloor x_\alpha \rfloor - 1$ and $s_\alpha = x_\alpha - \lfloor x_\alpha \rfloor$. The B_α's are the usual cubic B-spline basis polynomials. The problem of finding f is thus reduced to solving for the coefficients in Φ that approximate the scattered data points P.

We will defer to LWS for further details of the derivation, simply pointing out to the interested reader that their least-squares solution can be made immediately clear by considering a hyperplane in a 16-dimensional hyperspace.

The time and space complexity of this algorithm is $O(p + N^D)$, where p is the number of data points and N^D is the size of the control lattice.

There is a trade off between accuracy and smoothness with BA affected by the coarseness of the grid, which is a given. Because this algorithm uses cubic B-splines, the approximation function f is C^2 continuous and has local support. If we use a coarse control lattice, then the coefficients are calculated from many data points and f has wide support, so we get a smooth approximation. On the other hand, if we use a finer lattice, then f has smaller local supports and, hence, a smaller number of data points affecting the calculation of coefficients. Hence, f achieves greater accuracy.

To circumvent this situation, LWS went on to present the *multilevel* B-spline approximation, which they (and we) refer to as "MBA". This algorithm starts from the coarsest approximation (i.e., $N = 0$), which captures the global shape of f. By applying a finer lattice on the domain, a finer level of approximation can be obtained.

At the refinement stage, the function approximates the difference between the approximated data and the original data for the finer lattice. That is, given an initial coarsest fit f_0 to initial data y_c, it finds the approximation f_1 for $\delta^1 y = y_c - f_0(x_0, \ldots, x_{D-1})$.

Successively finer levels approximate and reduce the residual error. The final approximation is defined as the sum of all levels of fits f_k, i.e. $f = \sum_{k=0}^{h} f_k$. The only user-specified fit parameter is then the maximum level h, and with non-pathological data (i.e., all $\mathbf{x}_c$'s are numerically distinct), the least-squared error of the fit decreases monotonically with increasing h. This makes it possible for an MBA user to specify a given tolerance and have the algorithm find the smallest value of h which satisfies it.

Let N^D be the size of finest control lattice, p be the number of the data points, and h be the maximum level of refinement. Then the time complexity for the fitting part the MBA algorithm is $O(hp + N^D/(1 + 2^{-D}))$ and its space complexity is $O(p + N^D)$. LWS also showed how to subdivide and combine MBA levels to replace a hierarchy with a single mesh, making the evaluation of the fit independent of h (i.e. $O(1)$).

3 Fitting Measured BRDF Data

In this section, we describe the BRDF databases that we use, the fit to those data we construct, and our comparisons of our results to existing methods.

3.1 BRDF Databases

We use two measured BRDF on-line databases.

Foo [8] measured BRDF data using a custom-built gonioreflectometer and his data is available from the Cornell measurement web site[4]. They measured BRDFs for different wavelengths and each material has more than 1,300 samples. This database contains data for various types of paint. It also has data for human skin, sampled with a digital camera [20].

Dana et al. [7] used a robotic manipulator and CCD camera for their BRDF measurement. Their data is available from the CUReT database web site [5]. This database has reflectance measurements for over 60 different materials, each observed with over 200 different combinations of viewing and illumination directions. The data is multi-spectral, with red, green, and blue channels.

[1] f is a function of spatial coordinates directly. A parametric space, (e.g., (u, v)), is not involved.

Both the Cornell and the CUReT BRDF databases measure the BRDF for only half of the upper hemisphere, assuming bilateral symmetry. Hence the data we represent will not only be isotropic (discussed below), but symmetric: $f_r(\theta_i, \theta_o, \phi_o) = f_r(\theta_i, \theta_o, -\phi_o)$ [2].

3.2 BRDF Parameterization

In the case of an isotropic surface, the BRDF is invariant under rotation around the surface normal. Therefore, three parameters are enough to represent BRDF as $f_r(\theta_i, \theta_o, \delta\phi)$, where $\delta\phi = \phi_o - \phi_i$.

If we map the projection of the incident direction onto the tangent plane to the positive x axis, then $\phi_i = 0$ and $\delta\phi$ would be equal to ϕ_o. If the incident direction were not so aligned, then we can conceptually rotate the (isotropic) surface element around the surface normal until it is. Hence we may denote the BRDF as a function of three parameters: $f_r(\theta_i, \theta_o, \phi_o)$.

We could parameterize the BRDF strictly in terms of (possibly scaled) angles, but this would leave us with a polar anomaly: Our fitting procedure would not realize that $f_r(0, 0, \phi_o)$ refers to the same point for all ϕ_o. Any fit procedure could therefore produce undesirable artifacts near normal incidence and reflection. We need to adopt a coordinate system that avoids this anomaly.

One such, described by Lewis in [19], is "Nusselt coordinates". He first consider the x, y, and z direction cosines corresponding to a direction (θ, ϕ):

$$\mu_x = \sin\theta\cos\phi \quad \mu_y = \sin\theta\sin\phi \quad \mu_z = \cos\theta. \tag{5}$$

Then he affinely maps μ_x and μ_y to lie between 0 and 1:

$$\kappa = \frac{\mu_x + 1}{2} \quad \lambda = \frac{\mu_y + 1}{2} \tag{6}$$

These are the Nusselt coordinates. The mapping $(\theta_i, \theta_o, \phi_o) \leftrightarrow (\kappa_i, \kappa_o, \lambda_o)$ is unique and invertible. By applying this coordinate representation, a BRDF can be represented as $f_r(\kappa_i, \kappa_o, \lambda_o)$.

An added benefit of Nusselt coordinates is that μ_x and μ_y (and therefore κ_i, κ_o, and λ_o) are usually readily available as part of the scene geometry during rendering: no inverse trigonometry is required.

[2] This is different from Helmholtz reciprocity (cf. [18]), which would hold that $f_r(\theta_i, \phi_i, \theta_o, \phi_o) = f_r(\theta_o, \phi_o, \theta_i, \phi_i)$.

3.3 Fitting BRDF Data with the MBA Algorithm

Fitting either Cornell or CUReT data with the MBA algorithm turns out to be straightforward. Both databases use spherical coordinates, so we first convert them to Nusselt coordinates. We then apply a three-dimensional version of the MBA algorithm to the data.

Once we have the coefficients, to calculate the BRDF for a given $(\kappa_i, \kappa_o, \lambda_o)$, we need to map the Nusselt coordinate to the grid of coefficients (i, j, k). The control lattice is of dimension $(M+3)\times(M+3)\times(M+3)$, and for the MBA algorithm, $M = 2^h$ for the maximum level h. The gap Δ between each control point is $1/M = 2^{-h}$. Hence

$$i = \lfloor \kappa_i/\Delta \rfloor - 1 \quad s = \frac{\kappa_i}{\Delta} - \left\lfloor \frac{\kappa_i}{\Delta} \right\rfloor.$$

Similarly, m, t, n, and r can be determined from κ_o and λ_o, respectively. So at refinement level h, the fit is expressed as

$$f_r^h(\kappa_i, \kappa_o, \lambda_o) = \sum_{l=0}^{3}\sum_{m=0}^{3}\sum_{n=0}^{3} B_l(s)B_m(t)B_n(r)\phi^h_{(l+i)(m+j)(n+k)}, \tag{7}$$

where B_l's are bi-cubic B-spline basis functions and the ϕ^h's are the coefficients on the lattice of refinement level h, Φ_h.

3.4 Quantitative Results

We applied MBA algorithm as described previous subsection and measured some errors to show the accuracy of the fit.

We selected data for felt (No. 1), leather (No. 5), aluminum foil (No. 15) from the CUReT database and blue latex paint from the Cornell database. Felt shows a mostly-diffuse surface with slightly-specular behavior at high incident and reflected polar (grazing) angles . Leather has a little bit of specularity everywhere and high specularity at grazing angles. Aluminum foil has high specularity (See [5] for images of the sampled materials). Blue latex is largely diffuse but it also has a high specular peak at grazing angles.

Starting from an initial lattice size of 4 ($h = 0$), we increase the refinement level and compute error metrics at

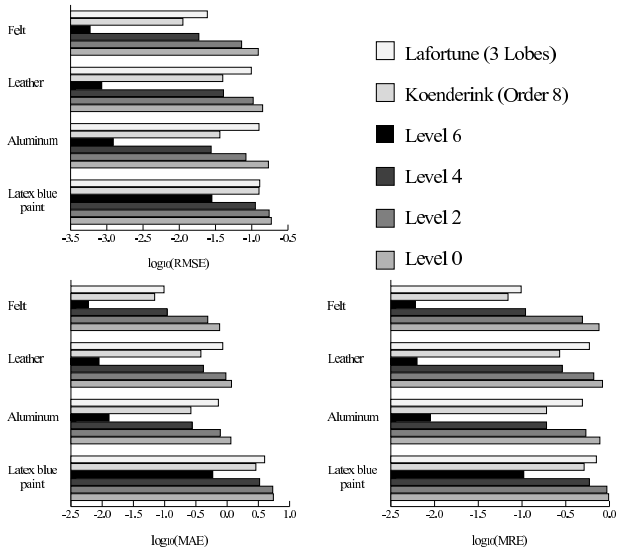


Figure 1: Error Measurements Compared with Lafortune and Koenderink Models.

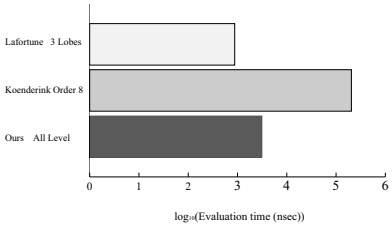


Figure 2: Comparison of Evaluation Time

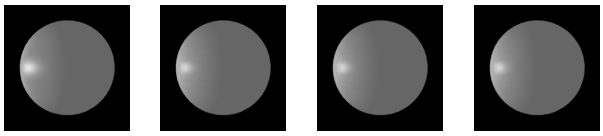


Figure 3: Varying Refinement Level (h) for Aluminum Foil. Light is coming from $60°$ to the left of the viewing direction.

each level. To illustrate storage requirements, we also tabulate the total number of coefficients at refinement level h, which is $(2^h + 3)^3$ (see LWS).

There is some disagreement on what constitutes a valid estimate of BRDF error, so to evaluate the fits quantitatively, we computed several. Results are shown in Figure 1. The first metric was the root mean square error ("RMSE" in Figure 1), which is computed as :

$$\text{RMSE} = \sqrt{\frac{\sum_{k=0}^{N-1} \left(y_k - f_r(\kappa_{ik}, \kappa_{ok}, \lambda_{ok})\right)^2}{N}} \qquad (8)$$

RMSE can be expected to measure how well a fit duplicates smooth, diffuse data, but often of more concern is how well a fit duplicates specular peaks, so we choose for a second metric the maximum absolute deviation from the measured data ("MAE" in Figure 1), which is defined as:

$$\text{MAE} = \text{MAX}_{k=0}^{N-1} \left| y_k - f_r(\kappa_{ik}, \kappa_{ok}, \lambda_{ok}) \right|. \qquad (9)$$

As (3) shows, BRDF values are dimensionless, but they are not unitless. (In this case, they have units of ster^{-1}.) So for a more intuitive grasp of the goodness of the fit, we also show a third metric, the maximum relative error, which is the maximum ratio of the absolute error to the maximum (absolute) sample datum ("MRE" in the Figure 1).

$$\text{MRE} = \frac{\text{MAX}_{k=0}^{N-1} \left| y_k - f_r(\kappa_{ik}, \kappa_{ok}, \lambda_{ok}) \right|}{\text{MAX}_{k=0}^{N-1} \left| y_k \right|}. \qquad (10)$$

To compare our results with those of other models, we chose two popular contemporary ones. The first is the one proposed by Koenderink et al.[13] and fit the measured data with order 8 of their method. The other is the one proposed by Lafortune et al.[14] with 3 cosine lobes. Hereafter, we will simply refer to these as "Koenderink method" and "Lafortune method", respectively.

In all cases, our level 6 shows greater accuracy than the other two by a factor of at least a factor of 4 ($\log_{10} 4 = 0.6$ in Figure 1), and usually an order of magnitude. Our level 4 (which uses much fewer coefficients than level 6) is comparable to the other two.

3.5 Speed of Evaluation

As the computation of the fit can presumably be done prior to rendering, it is irrelevant to rendering speed. The

fit evaluation time is therefore the critical one, so we have measured it and compared it with those of the Koenderink and Lafortune models. We measured the mean time of 100,000 function calls for different combinations of incident and viewing direction. The evaluations were done on a PC running Linux RedHat 7.2 on an AMD Athlon 1.4Ghz microprocessor with 256 KB cache and 256 MB system memory. The results are listed in Figure 2.

Generally, representations with summation or products of basis functions must increase the number of terms to obtain greater accuracy. This leads to an increase in the evaluation time as well, which can be observed with the other methods. On the other hand, the time required for our MBA-based method doesn't increase with the accuracy of the fit: We still evaluate a $4 \times 4 \times 4$ submesh of the grid.

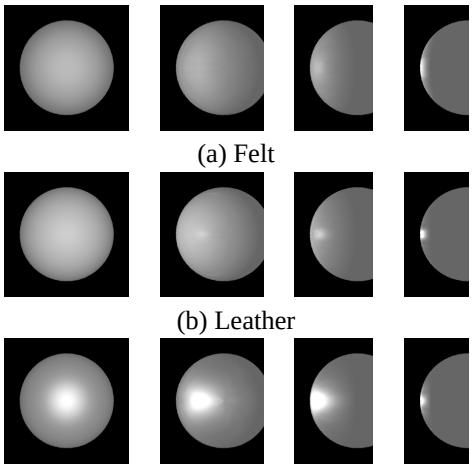
3.6 Visual Results

Figure 3 shows the synthesized sphere using BRDFs fit by our method for different refinement levels. As the refinement level increases, the specular region narrows and its brightness increases. This shows that the fit improves as the refinement level increases. However, even in level 3 (with 1,331 coefficients), the fit can capture the characteristics of the surface well. So, if we can sacrifice some accuracy, we can synthesize a fairly reasonable image with less storage.

Figure 4 shows a synthesized sphere for different materials with refinement level 6. The image shows the smooth transition from specular to diffuse and all images show the expected characteristics of each material. With an ad-hoc BRDF model, specularity at grazing angles is difficult to represent, but our model can correctly represent such specular features, since measured data captures values at such angles and the fit closely matches that data.

3.7 Storage Requirements

At refinement level 6, the root mean square error and maximum deviation get small enough to say that the fit function matches all the data well. Although the errors at this level are negligible, the storage requirements are not. Figure 5 shows the storage requirement (number of parameters) comparison of our method and other methods.



(a) Felt

(b) Leather

(c) Aluminum Foil

Figure 4: Varying Materials and Light Directions. From left to right, light is coming from 0° (i. e., coincident), 30°, 60° and 90° to the left of the viewing direction.

If we use single precision (4 bytes) to represent a coefficient, we need about 1.2MB of storage to represent the fit function. This applies to a single spectral channel, so this number would be tripled for the three RGB coefficients in a practical application.

Our method requires far more storage than others. While it doesn't increase the evaluation time of fit, this storage requirement is a major disadvantage of the method as we have described it so far.

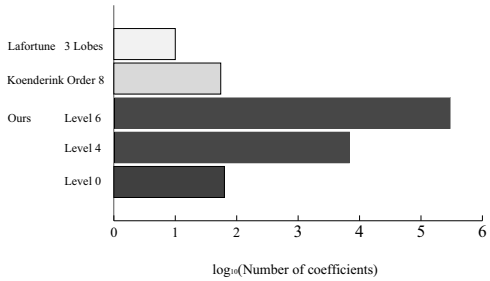


Figure 5: Storage comparison with other methods

4 Compression

In this section we describe how we can reduce the number of coefficients without sacrificing the accuracy of the fit.

4.1 Decomposing the BRDF into Two Parts

A highly specular material reflects light mostly in the reflected direction. Therefore, coefficients that describe regions away from the reflected direction are quite small. The convex hull property of the B-spline representation would allow us to "discard" small coefficients, effectively replacing them with zero, without adverse effects. This would suggest a sparse representation, storing only coefficients above a certain threshold and assuming the rest were zero.

On the other hand, a highly diffuse material can be described well at a relatively low level of refinement, i.e., with a small number of coefficients. The materials we are trying to represent have properties that range from highly specular to highly diffuse, so a combined representation suggests itself.

We partition the fit into two separate fits, referring to them as the diffuse fit f_d and the specular fit f_s. f_d is done with a coarse mesh and is non-sparse. f_s uses a fine mesh and is sparse. The BRDF is then the sum of two fits $f_r = f_d + f_s$.

The construction of the two fits is straightforward, and resembles that of the MBA algorithm itself: We construct f_d first and then f_s is fit to the differences between the data and f_d.

An alternative we considered and rejected was described in the LWS paper: a hierarchy of sparse fits. The more sparse fits we have, the longer the evaluation time required[3], and it takes longer to look up the sparse coefficients on a level than it does to simply index the non-sparse coefficients.

4.2 Technique

A number of choices are available for sparse storage of f_s. We chose to use minimal perfect hashing [6], as implemented by Jenkins [11]. BRDF coefficient storage is ideally suited to this technique, as the database is completely known at the time of hash table construction and the time required to construct the table is a secondary consideration at best.

[3]This would incur the same penalty as that of orthogonal basis function expansions.

	Compression	
	90 %	95 %
Diffuse Level 3		
Number of CPs	31,407	16,369
Total Storage(Bytes)	426,647	366,495
Diffuse Level 4		
Number of CPs	36,935	21,897
Total Storage(Bytes)	448,759	388,607

Table 1: Storage Required After Compression. In all cases, the specular fit was level 6 with single precision floating point.

Table 1 shows the number of coefficients and storage requirements for various bi-level fits. Total storage in the table includes the coefficients themselves as well as the overhead required for hashing.

4.3 Results

We applied our compression method to the same BRDF database as before and measured storage requirement and errors. In addition, we synthesized spheres with the compressed fit.

4.3.1 Quantitative Results

Figure 6 shows plots of our fit with various combinations of levels and compression percentages contrasted with the Lafortune and Koenderink methods. We use (D,S,C) to refer the combination of diffuse level(D), specular level(S) and compression percentage(C) for the fit. For example "(3,6,90)" is the fit using level 3 diffuse and level 6 specular fit and 90% of the coefficients for the specular fit are omitted.

Even with a diffuse level of 1, our method can achieve error levels comparable to the other two. Models (3,6,90) and (4,6,95) compare very favorably. Although (1,6,90) is in almost all cases as good as or better than Koenderink or Lafortune, it suffers in comparison to (3,6,90)

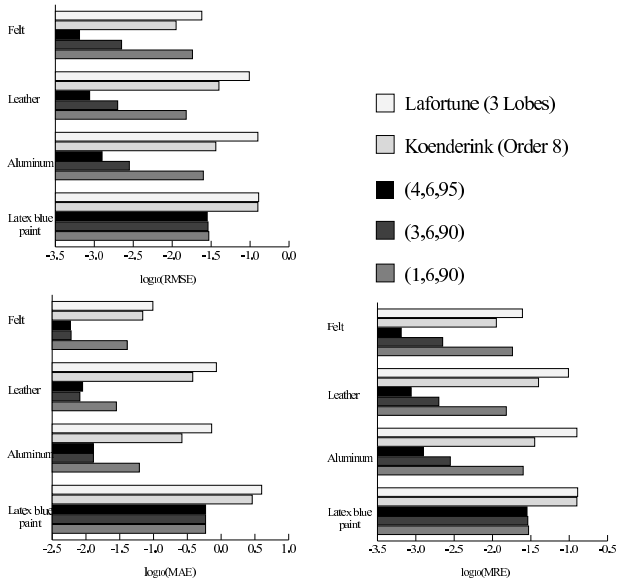


Figure 6: Error Measurements With Compression Compared to Lafortune and Koenderink Models.

and (4,6,95) because diffuse level 1 just does not capture f_d adequately, leaving it to be captured by the f_s fit, and when compression removes f_s coefficients, the results suffer.

One feature not shown in these figures is the presence of "ringing": oscillations that are observed in the fit in regions away from the data points. Similar in nature to "Gibbs overshoot" observed in Fourier reconstruction, they are a result of the mismatch between sampling (spatial) frequency and the resolution of the fit function (in this case, the grid spacing). We have found that ringing can be eliminated by increasing the size of the diffuse component. As even compressed specular data dominates the amount of required storage, this does not serious affect the storage requirement. See Kameya [2] for further details.

4.3.2 Visual Results

Figure 7 shows spheres rendered with a compressed fit. All spheres show a smooth transition from specular to diffuse reflection. The plot shows ringing, however, in the (1,6,90) fit.

Finally, in Figure 8 we render a more complex object with compressed (4,6,95) and non-compressed (level 6) BRDFs. The differences within each pair are undetectable, leading us to the conclusion that compression is a viable means of reducing storage overhead at no sacrifice to image quality.

5 Conclusion and Future Directions

In this paper, we presented a smooth representation for measured BRDFs by using a non-parametric multi-level B-spline approximation to fit reflectance data. The resulting fit function approximates the the dataset well. By increasing the size of the control lattice, the algorithm finds an arbitrarily good fit to the data.

One drawback compared to other methods is the amount of storage required. We addressed this by devising a compression scheme which we showed could substantially reduce storage needs with no visible impairment in image quality.

As future work, we hope to take better advantage of the MBA algorithm's arbitrary dimensionality and extend this work to anisotropic surfaces, subsurface scattering (cf. [12]), and texture representation. Furthermore, the algorithm should lend itself quite well to implementation on programmable shading hardware.

References

[1] J. F. Blinn. Models of light reflection for computer synthesized pictures. *Computer Graphics*, 11(2):192–198, July 1977.

[2] Co-Author. *A Smooth, Efficient Representation of Reflectance*. PhD thesis, Institution Omitted.

[3] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, January 1982.

[4] Cornell reflectance data, http://www.graphics.cornell.edu/online/-measurements/reflectance/index.html.

[5] Curet reflectance and texture database, http://www.cs.columbia.edu/cave/curet/.

[6] Z. J. Czech, G. Havas, and B. S. Majewski. Perfect hashing. *Theoretical Computer Science*, 182, 1997.

[7] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. Technical report, Columbia University, December 1996.

[8] S. Foo. A gonioreflectometer for measuring the bidirectional reflectance of materials for use in illumination computations. Master's thesis, Cornell University, Ithaca, New York, July 1997.

[9] A. Fournier. Separating reflection functions for linear radiosity. In *Eurographics Rendering Workshop 1995*, pages 296–305, Dublin, Ireland, June 1995. Eurographics.

[10] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):175–186, July 1991.

[11] B. Jenkins. Algorithm alley : Hash functions. *Dr. Dobb's Journal*, September 1997.

[12] H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. *Proceedings of SIGGRAPH 2001*, pages 511–518, August 2001.

[13] J. J. Koenderink, A. J. van Doorn, and M. Stavridi. Bidirectional reflection distribution function expressed in terms of surface scattering modes. *Proc. 4th European Conference on Computer Vision*, pages 28–39, 1996.

[14] E. P. F. Lafortune, S. C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. *Proceedings of SIGGRAPH 97*, pages 117–126, August 1997.

[15] J. Lambert. Photometria sive de mensure de gratibus luminis, colorum umbrae. *Eberhard Klett*, 1760.

[16] G. J. W. Larson. Measuring and modeling anisotropic reflection. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):265–272, July 1992.

[17] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel b-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, July - September 1997.

[18] R. R. Lewis. Making shaders more physically plausible. *Computer Graphics Forum*, 13(2):109–120, January 1994.

[19] R. R. Lewis and A. Fournier. Wavelet radiative transfer and surface interaction. *Computer Graphics Forum*, 19(2):135–151, June 2000.

[20] S. Marschner, S. Westin, E. Lafortune, K. Torrance, and D. Greenberg. Image-based brdf measurement including human skin. *Eurographics Rendering Workshop 1999*, June 1999.

[21] M. D. McCool, J. Ang, and A. Ahmad. Homomorphic factorization of brdfs for high-performance rendering. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, pages 171–178. ACM Press / ACM SIGGRAPH, August 2001. ISBN 1-58113-292-1.

[22] M. Oren and S. Nayar. Generalization of the lambertian model and implications for macine vision. *International Journal of Computer Vision*, 14:227–251, 1995.

[23] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

[24] C. Schlick. A customizable reflectance model for everyday rendering. *Fourth Eurographics Workshop on Rendering*, pages 73–84, June 1993.

[25] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In *Proceedings of SIGGRAPH 95*, Computer Graphics Proceedings, Annual Conference Series, pages 161–172, Los Angeles, California, August 1995. ACM SIGGRAPH / Addison Wesley. ISBN 0-201-84776-0.

[26] F. X. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg. A global illumination solution for general reflectance distributions. *Computer Graphics (Proceedings of SIGGRAPH 91)*, 25(4):187–196, July 1991.

[27] S. H. Westin, J. R. Arvo, and K. E. Torrance. Predicting reflectance functions from complex surfaces. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26(2):255–264, July 1992.

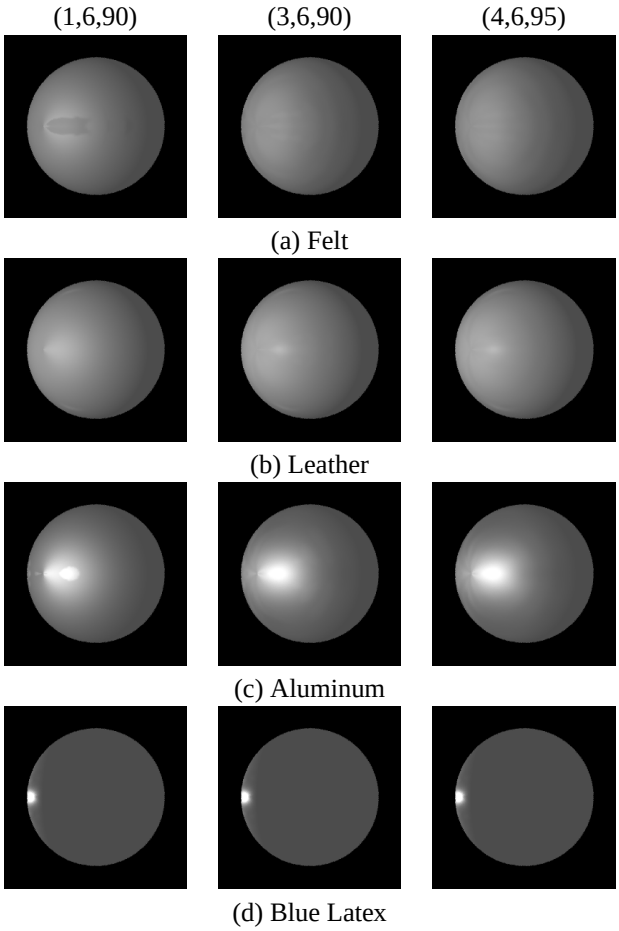
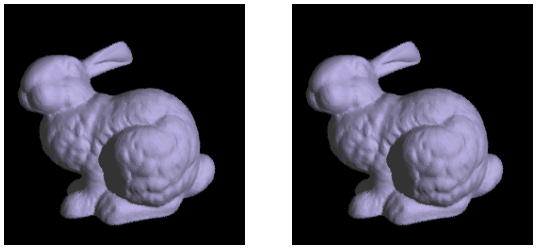


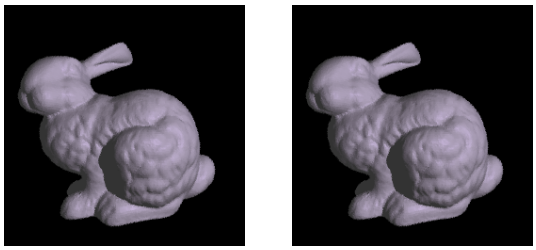
Figure 7: Spheres Rendered with Compressed Fit. Spheres on the top three rows are lit from 30° to the left of the viewer and those on the bottom row are lit from 90° to the left of the viewer.

Felt



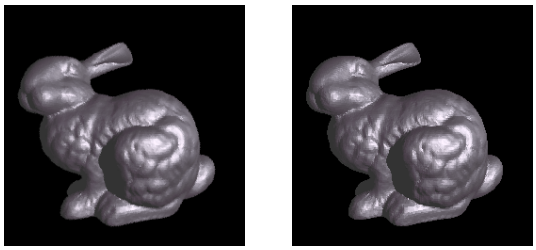
No compression(Level 6) (4,6,95)

Leather



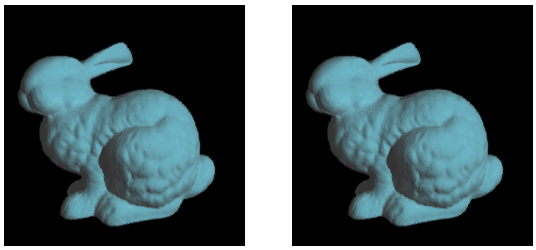
No compression(Level 6) (4,6,95)

Aluminum



No compression(Level 6) (4,6,95)

Blue Latex



No compression(Level 6) (4,6,95)

Figure 8: Synthesized Bunny Without and With Compression