A Wavelet Representation of Reflectance Functions

Paul Lalonde and Alain Fournier

August 26, 1997

Abstract

Analytical models of light reflection are in common use in computer graphics. However, models based on measured reflectance data promise increased realism by making it possible to simulate many more types of surfaces to a greater level of accuracy than with analytical models. They also require less expert knowledge about the illumination models and their parameters. There are a number of hurdles to using measured reflectance functions, however. The data sets are very large. A reflectance distribution function sampled at five degrees angular resolution, arguably sparse enough to miss highlights and other high frequency effects, can easily require over a million samples, which in turn amount to over 4 megabytes of data. These data then also require some form of interpolation and filtering to be used effectively.

In this paper we examine issues of representation of measured reflectance distribution functions. In particular we examine a wavelet basis representation of reflectance functions, and the algorithms required for efficient point-wise reconstruction of the BRDF. We show that the non-standard wavelet decomposition leads to considerably more efficient algorithms than the standard wavelet decomposition. We also show that threshholding allows considerable improvement in running times, without unduly sacrificing image quality.

1 Introduction

One of the goals of image synthesis is to produce images of three dimensional computer generated scenes that are indistinguishable from images of real scenes. The most successful methods used to render such images have been based on simulating the physics of light transport [3, 14]. The process starts with a model of the scene to be rendered, including descriptions of light sources and surface reflection properties, and proceeds to simulate the transport of light from sources to surfaces and from surface to surface.

At their lowest level, most rendering algorithms depend on models of local illumination to define the interplay of light with surfaces. These models describe how light is scattered by a surface through reflection and transmission. They range from ad hoc analytical models, to empirically based physical models, through to representations of data measured from physical samples.

Ray tracing methods consider only direct illumination and perfect reflections and refraction [27], although extensions exist to model other phenomena [14, 25, 22, 10]. Radiosity methods consider the inter-reflections of light in scenes with only diffuse surfaces [3]. Again there are extensions for other lighting phenomena [11, 21, 7]. Both methods use simple models of light sources and local reflection.

As the sophistication of rendering methods has increased, traditional reflectance models such as those presented by Phong [19], Blinn [1], and Cook and Torrance [4] have become less adequate to the task at hand. Many extensions to these methods have been proposed that make use of more complex lighting and reflection models. However, little progress has been made in the use of reflectance distributions obtained through empirical measurement or simulation.

The main reasons for this lack of progress are that the data are expensive to acquire, that the data sets are quite large, causing major difficulties even on computers with large memories, and that most rendering techniques cannot use a direct formulation of the BRDF.

We propose to use a wavelet representation of a measured BRDF that is both compact, allows a trade-off of storage size to accuracy, and provides a simple interpolation mechanism for the original BRDF samples. In other work we show how this representation can be used to generate reflected directions for Monte-Carlo ray tracing [17], and how to use the wavelet representation to efficiently perform shading computations filtered over the incident hemisphere [16].

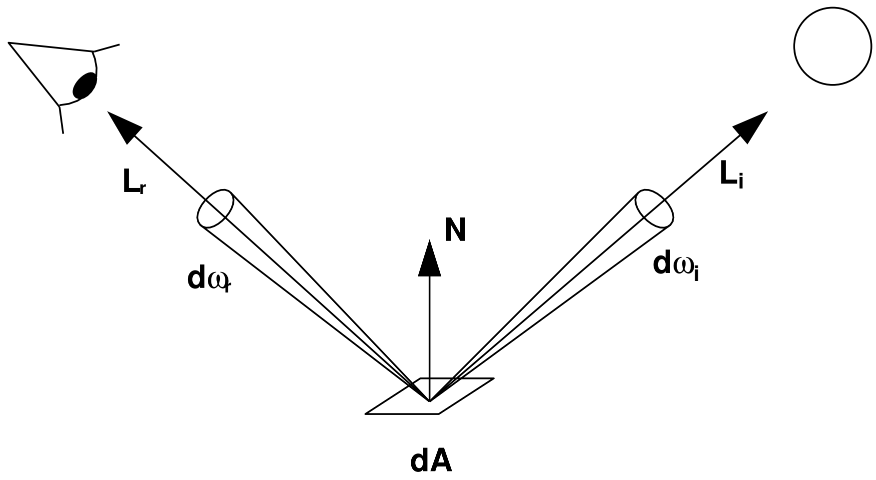


Figure 1: Shading Geometry

2 The Bidirectional Reflectance Distribution Function

The bidirectional reflectance distribution function (BRDF) characterizes the reflection of light at a surface [12], assuming that there is no subsurface scattering. The BRDF gives the ratio of incident light from a particular solid angle to reflected light at another solid angle at all wavelengths[1].

More formally, consider the geometry in Figure 1. When light arrives at a differential area dA through an incident solid angle [2] $d\omega_i$, in direction $\vec{\omega_i}$, the amount reflected in some other direction $\vec{\omega_r}$ from dA is proportional to the incident irradiance from $\vec{\omega_i}$:

$$dL_r(\vec{\omega_r}) \propto dE(\vec{\omega_i}), \tag{1}$$

which is to say that an increase in incident light energy per unit area will lead to a proportional increase in the reflected energy. If we include the dependence on wavelength λ, Equation 1 can be written as

$$dL_r(\vec{\omega_r}, \lambda) \propto dE(\vec{\omega_i}, \lambda) \tag{2}$$

The meaning now changes slightly: when light arrives at dA through $d\omega_i$, the amount of that light reflected in some other direction $\vec{\omega_r}$ at wavelength λ is proportional to the incident irradiance from $\vec{\omega_i}$ at wavelength λ The constant of proportionality in Equation 2 is the BRDF [3]. Rewriting and substituting the change in irradiance as a result of the illumination of the patch from $d\omega_i$:

$$dE(\vec{\omega_i}, \lambda) = L_i(\vec{\omega_i}, \lambda) \cos\theta_i d\omega_i \tag{3}$$

we find the BRDF f_r:

$$\mathcal{F}_r(\vec{\omega_i} \rightarrow \vec{\omega_r}, \lambda) = \frac{L_r(\vec{\omega_r}, \lambda)}{L_i(\vec{\omega_i}, \lambda) \cos\theta_i d\omega_i}$$

The dependence on four angles is frequently made explicit by writing the BRDF as $\mathcal{F}_r(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)$.

2.1 Measured Reflectance Functions

Measuring a BRDF from an actual surface is problematic for several reasons. A gonioreflectometer measures a BRDF by physically moving a light source and a radiometer over the hemispheres of incident

[1] The bidirectional transmission distribution function (BTDF) similarly describes the transmission of light through a surface. The union of the BRDF and BTDF, the bidirectional transmission and reflectance distribution function (BTRDF) describes both reflection and transmission at a point on a surface. Most of our discussion will refer to the BRDF, but the results are symmetrically applicable to BTDFs and so also to BTRDFs.

[2] Differential solid angles have associated with them a unique direction. When we write $\vec{\omega}$ we are refering to the direction associated with the differential solid angle $d\omega$.

[3] The ANSI/IES standard defines reflection as "the process by which the incident flux leaves a [surface] from the incident side, without change in frequency"[4].Some reflection-like processes do cause a change in frequency but these will not be addressed here. The extensions to do so are straightforward. Likewise, these expressions can be written to account for polarization of light. Again, these will not be examined here.

and reflected directions, taking measurements every few degrees. Gonioreflectometers are relatively expensive and slow, and the data returned from them is often quite noisy, both in position and in value. Ward attempted to address the problems of cost and speed [24] but the samples obtained from his gonioreflectometer are both nonuniform and noisy, complicating their use in rendering. Ward overcomes these limitations by fitting the parameters of a simple anisotropic shader to his data and using this shader to render surfaces, but his images are limited by his shading model.

It is also possible to simulate a gonioreflectometer in software. Using a ray-tracing strategy, a micro-geometry representing the small scale surface properties of a material is sampled [13, 2, 26, 8, 6]. These samples are then tabulated and can be used in the same way as a physically measured BRDF can. For research purposes these synthetic BRDFs have a number of advantages. The noise and errors can be simulated, and samples can be taken at specific points without having to deal with the inaccuracies involved with mechanical systems. The sampling is both fast and repeatable.

Once measured and tabulated, BRDF data sets are large enough that their size is an impediment to their use. Without a priori knowledge of a material being sampled with a gonioreflectometer, or of the properties of a micro-geometry being sampled by a virtual gonioreflectometer, it is not possible to establish what sampling density is required to represent the BRDF within a given error tolerance. Either a relatively low frequency sampling is performed, possibly missing important features such as sharp reflection peaks, or a high frequency sampling is used, resulting in a data set which is too large to be useful. Consider for example a Phong model with an exponent of 50. It's specular component falls by half in 9.6 degrees. Sampling every 10 degrees gives $(36 * 9)^2 = 104976$ samples. Each of these may involve multiple spectral samples.

2.2 Parameterization

When sampling or encoding the BRDF some parameterization of the incident directions is required, whether it be simply by polar and azimuthal angles, or something more sophisticated. We choose a variant of the Nusselt embedding, which is related to direction cosines [18].

To express a direction given by a unit vector $\vec{v}$ in a local coordinate frame given by the normal $\vec{N}$ and a surface tangent $\vec{T}$, we let $\mu_x = \vec{v}.\vec{T}$ and $\mu_y = \vec{v}.(\vec{N} \times \vec{T})$. We normalize these to the range (0..1), and call them κ and λ:

$$\kappa = (\mu_x + 1.0)/2.0$$
$$\lambda = (\mu_y + 1.0)/2.0$$

κ and λ then index in the unit square. Directions then cover only the circle at the center of the square. We define the value of any function of directions for which $\mu_x^2 + \mu_y^2 > 1$ as zero.

This parameterization has one main advantage. The cosine term for projected area that appears in local shading equations is absorbed by the parameterization. If we take for example the local shading equation, omitting possible emission of the surface, in terms of polar angles and where $L_i()$ is the incident radiance and$\mathcal{F}_r$ is the BRDF:

$$L_r(x, y, \theta_r, \phi_r) = \int_\theta \int_\phi L_i(x, y, \theta, \phi)\mathcal{F}_r(\theta, \phi, \theta_r, \phi_r) \cos\theta d\omega$$

To perform the change of coordinates we use the determinant of the Jacobian of the change of variables which is:

$$\left|\frac{\partial(\theta, \phi)}{\partial(\kappa, \lambda)}\right| = \frac{4}{\cos\theta \sin\theta}.$$

In Nusselt coordinates, recalling that $d\omega = \sin\theta d\theta d\phi$ we get:

$$L_r(u, v, \kappa_r, \lambda_r) = 4 \int_0^1 \int_0^1 L_i(u, v, \kappa, \lambda)\mathcal{F}_r(\kappa, \lambda, \kappa_r, \lambda_r) d\kappa d\lambda \quad (4)$$

Thus we write the BRDF as $\mathcal{F}_r(\kappa_i, \lambda_i, \kappa_r, \lambda_r)$. A disadvantage of this parameterization is that since we do not stretch the circle to the unit square the corners are filled with zeros, as described above. The effect of thisis that after thresholding the wavelet coefficients of a function with such a parameterization non-zero values may appear outside the unit disk. Though this may lead to small violations of energy conservation, no visual artifacts are produced since these values are never used.

2.3 Using Measured BRDFs

The most naive approach to using a tabulated BRDF is to use a piecewise constant reconstruction strategy. This technique, though easy to implement and computationally efficient, is usually insufficient for even the simplest BRDFs. The lack of continuity in the representation causes highly visible artifacts in the generated images. Quadrilinear interpolation on the parameters $\phi_i, \theta_i, \phi_r, \theta_r$, provides geometric $C^{[0]}$ continuity, but the lack of $C^{[1]}$ tangential continuity leads to visible artifacts (Figure 2(a)). Higher order interpolation functions can provide us with $C^{[1]}$ tangential continuity. We used a tensor product quadri-cubic B-Spline scheme to interpolate the BRDF and found the approximation to be quite good (Figure 2(b)). The main difficulty is that 256 control vertices need to be examined, making the evaluation quite expensive. Also, both methods require the full data set to be stored in memory. This quickly gets prohibitive, as the data sets are about four megabytes in size.

3 A Wavelet Representation

The raw tabulated representations of the BRDF is impractical because of the large size of the data set. Instead we want to find a space efficient representation that is faithful to the original data and quick to use. The method has to be competitive in speed with analytic and algorithmic shaders. In addition the method must provide interpolation of the original data at low cost.

We propose a wavelet based solution for a number of reasons:

- the representation allows a good control of the accuracy/space trade-off;
- the point-wise reconstruction is logarithmic in the number of original BRDF samples;
- a wavelet reconstruction can smoothly interpolate the original data;
- some error metrics are easy to compute;
- incremental levels of accuracy can be used; and
- BRDFs are largely smooth with important discontinuities, which can be interpreted as a signal which is mostly low-frequency, but with localized high frequencies. Wavelets are idealy suited to represent this kind of data.

3.1 Wavelets and BRDFs

Schröder and Sweldens showed a method for encoding the reflections from one incident direction using a spherical wavelet representation [20]. This restriction to one incident direction is a serious shortcomming. This method does not easily extend to encoding a large number of incident directions since the dimensionality of the function space of a BRDF does not map to the topology of their spherical space.

We chose instead to use a multi-dimensional wavelet transform, mapping the parameter space of the BRDF to a cartesian grid. The multi-dimensional wavelet decomposition allows a choice in how the uni-dimensional decomposition is extended to higher dimensions. In the standard case a product of one-dimensional decompositions is used. In the non-stardard case a product of the basis functions is used, leading to a multi-dimensional multi-resolution analysis that is analogous to the uni-dimensional case[5]. We will examine each in turn, and the algorithms induced.

3.2 Standard Decomposition

The standard four-dimensional wavelet decomposition yields:

$$\mathcal{F}_r(\kappa_i, \lambda_i, \kappa_r, \lambda_r) = \sum_g \sum_h \sum_j \sum_k c_{g,h,j,k} B_g(\kappa_i) B_h(\lambda_i) B_j(\kappa_r) B_k(\lambda_r) \tag{5}$$

where B_n is defined as

$$B_n(x) = \begin{cases} \phi(x) & \text{if } n = 0 \\ 2^{-l/2}\psi(2^{-l}x - m) & \text{if } n = 2^l + m \text{ for some } l \text{ and } m \geq 0 \end{cases} \tag{6}$$

and ψ is the mother wavelet and ϕ is the smoothing function.

3.3 Non-Standard Decomposition

To manage the notation for the non-standard multidimensional wavelet bases, we will use a four-variable vector $\mathbf{d}$:

$$\mathbf{d} = (\kappa_i, \lambda_i, \kappa_r, \lambda_r)$$

and a multi-resolution index $\mathbf{j}$:

$$\mathbf{j} = (\nu, l, m_{\kappa_i}, m_{\lambda_i}, m_{\kappa_r}, m_{\lambda_r})$$

Here ν is the *selector index* for the wavelet functions, l is the level in the wavelet subspaces, 0 being the coarsest, and m_p the offset at that level for parameter p. The level is the same for all variables because we use non-standard multidimensional wavelet bases [5, 9, 23]. Since the bases corresponding to different ν values all have the same hyper-cube of support (same extent for all variables at a given level) the data structure is simpler, as are many geometric operations. The selector ν determines the product of wavelet and smooth basis functions that together define $D_{\mathbf{j}}(\mathbf{d})$. We use a special notation to indicate the selection, where $\nu\{h\}$ is interpreted as the hth bit of ν:

$$\Gamma^h_{l,m,\nu}(p) = \left\{ \begin{array}{ll} \phi_{l,m}(p) & \text{if } \nu\{h\} = 0 \\ \psi_{l,m}(p) & \text{if } \nu\{h\} = 1 \end{array} \right\}$$

Then we can write the basis function $D_j(\mathbf{d})$ as:

$$\begin{aligned} D_j(\mathbf{d}) &= \Gamma^0_{l,m_{\kappa_i},\nu}(\kappa_i) \times \Gamma^1_{l,m_{\lambda_i},\nu}(\lambda_i) \times \\ & \quad \Gamma^2_{l,m_{\kappa_r},\nu}(\kappa_r) \times \Gamma^3_{l,m_{\lambda_r},\nu}(\lambda_r) \end{aligned} \tag{7}$$

We then express the BRDF $\mathcal{F}_r$ as its projection onto a multi-resolution wavelet space:

$$\mathcal{F}_r(\kappa_i, \lambda_i, \kappa_r, \lambda_r) = \sum_j f_{\mathbf{j}} D_{\mathbf{j}}(\kappa_i, \lambda_i, \kappa_r, \lambda_r) \tag{8}$$

where $D_{\mathbf{j}}()$ are the appropriate basis functions, indexed by $\mathbf{j}$, and $\tilde{D}_{\mathbf{j}}()$ are their duals. Then the $f_{\mathbf{j}}$ are given by

$$\begin{aligned} f_{\mathbf{j}} &= < \mathcal{F}_r(\kappa_i, \lambda_i, \kappa_r, \lambda_r) | \tilde{D}_{\mathbf{j}}(\kappa_i, \lambda_i, \kappa_r, \lambda_r) > \\ &= \int_0^1 \int_0^1 \int_0^1 \int_0^1 \mathcal{F}_r(\kappa_i, \lambda_i, \kappa_r, \lambda_r) \tilde{D}_{\mathbf{j}}(\kappa_i, \lambda_i, \kappa_r, \lambda_r) d\kappa_i d\lambda_i d\kappa_r d\lambda_r \end{aligned} \tag{9}$$

Since the BRDF is defined as zero when $\mu_x^2 + \mu_y^2 >= 1$, which corresponds to a range of κ and λ of between zero and one, we can safely narrow the bounds of integration from $(-\infty..\infty)$ to $(0..1)$.

3.4 Point Reconstruction

The difficulty with wavelet decompositions of the BRDF is that although it is efficient to reconstruct the entire original BRDF from the wavelet coefficients using the fast wavelet transform we can not afford the space or time to do so while rendering, unless we know that most incident and reflected directions and combinations thereof will be used, which is unlikely. Instead we want the BRDF only for specific incident and reflected angles. Likewise we would prefer not to examine every non-zero coefficient during the evaluation step. When reconstructing the BRDF at a point $\mathbf{d} = (\kappa_i, \lambda_i, \kappa_r, \lambda_r)$ we only want to examine basis functions whose supports overlap $\mathbf{d}$. The expectation is that when the basis functions have narrow support this will be considerably fewer terms than in the whole transform.

3.4.1 Standard Decomposition

Consider the coefficients and basis functions that must be examined to reconstruct the BRDF at point $\mathbf{d}$ in the standard decomposition. Each coefficient $c_{g,h,j,k}$ for which any of $B_g(\kappa_i), B_h(\kappa_i), B_j(\kappa_i), B_k(\kappa_i)$ is non-zero needs to be examined. Recall that $B_n(x) = 2^{-l/2}\psi(2^{-l}x - m)$ if $n = 2^l + m$ for some l and $m \geq 0$ when $n > 0$. We can then, for each l and m, establish which functions $\psi_{l,m}(x)$ are non-zero. If the wavelet has a width of support w then there will be w basis functions to examine at each level l. Observe then that each term the summations of equation 5 can reference basis functions $\psi_{l,m}()$ from different levels. This interaction means that a reconstruction costs $O(w^4 l^4)$; if there are n^4 original samples in the signal the cost is $O(w^4 \log_2^4 n)$.

3.4.2 Non-Standard Decomposition

The time complexity is better for the non-standard decomposition. Since the basis functions $D_{\mathbf{j}}()$ used in the analysis are multi-dimensional, each of the terms of the summation in equation 8 has a unique level l for all combinations of ψ and ϕ required to assemble $D_{\mathbf{j}}()$. There are then 16 basis functions with w^4 translations at each level that support the point $\mathbf{d}$ at which we are reconstructing, where w is the width of the basis functions. This means we have only $O(w^4 \log_2 n)$ terms to examine during our traversal to evaluate at a point. Another advantage of non-standard decomposition is that there are efficient algorithms for evaluating area integrals over regions of the function domain. This is examined in more detail with regards to generating random deviates from BRDF data [17].

In view of these results, we choose to use only the non-standard decomposition in our investigations.

3.5 Choice of Basis

Our choice of wavelet basis functions is determined by two properties: the width of support of the reconstruction basis, and the amount of compression a basis can provide. Since one of our goals in using the wavelet representation is to reduce the memory requirements we see immediately why greater compression is a goal. It is not so obvious why the width of support of the basis is important, or how it relates to compression rates.

Firstly, the narrower the width of support of the reconstruction wavelet, the less computational work is required in the reconstruction, since the scaled and translated wavelets will cover fewer terms. The dual basis functions, used for analysis, can be as wide as necessary since we can use the fast wavelet transform and perform the analysis offline. When extending to higher dimensions the amount of work required by wider bases increases with the power of the dimension. The number of terms examined then becomes a dominant factor in the cost of the reproduction. In the case of bi-orthogonal wavelets all we care about is the width of the reconstruction wavelet. The analysis wavelet can be as wide as necessary, since the wavelet transform is performed off-line using the fast wavelet transform.

Secondly, wider bases tend to compress better. The number of vanishing moments of a wavelet tells us the highest degree polynomial that that wavelet basis can represent exactly with only as many terms as the width of the basis. This ability to represent high-order polynomials correlates strongly with the amount of compression that a basis provides. In general, wavelets with more vanishing moments have a wider width of support. Thus, wider bases provide better compression, but at the cost of much more expensive computation.

It follows then that the best bases for our application will compress the data well, while being as narrow as possible. It is only possible to determine experimentally which bases are best for our application.

3.6 Data Structures

The remaining difficulty is in representing the coefficients. The major advantage of using a wavelet based representation is that the representation is sparse, with many of the wavelet coefficients being zero or small. To increase our compression rate at the cost of accuracy we threshold small coefficients to zero. A property of the wavelet transform is that the sum of these thresholded coefficients gives us the root mean square of the error induced by the thresholding. If our sigal is comprized of N data sample $\{f_i\}$ and $\{\tilde{f}_i\}$ are the corresponding values of the signal reconstructed excluing the k smallest coefficients $d_j, j = 1..k$ of our wavelet transform of f then:

$$\sum_{i=1}^{N}(f_i - \tilde{f}_i)^2 = \sum_{j=0}^{k} d_j^2 \tag{10}$$

The difficulty is that there are as many wavelet coefficients as original data points, even though a large number of them are zero. Thus we want to find a representation that stores only the non-zero coefficients, allowing us considerable space savings.

In a one dimensional transform it is possible to use a binary tree structure that at each node records the value of the coefficient and mimics the structure of the wavelet analysis. Then it is possible to prune any subtrees that contain only zero coefficients. There may remain nodes in the tree that record zero values (if any of its descendants are non-zero) but these add no more than a small linear factor to the size of the stored tree. For reconstruction it is sufficient to traverse a path down the tree, following a child pointer if the child's basis function supports the point being evaluated.

An efficient option for the representation is as a *Wavelet Coefficient Tree*. This tree is a sparse hexadecary tree (16 ordered children), where the depth of the nodes indicates the depth of the coefficients in the representation, and where each node stores the non-zero coefficients with the same offset m, indexed by ν, the wavelet basis selector.

We use a structure called a *wavelet node* that stores all non-zero wavelet coefficients with the same indices (l, m_0, m_1, m_2, m_3), and different basis selector ν. The wavelet node also maintains a list of non-empty subtrees rooted at a wavelet node with indices $l+1, 2m_0+b\{0\}, 2m_1+b\{1\}, 2m_2+b\{2\}, 2m_3+b\{3\}$ where $b = 0..15$. The addition of 2 16-bit masks indicating which coefficients (the value mask) and which children (the child mask) are present allow very compact storage of the tree.

This method is both simple to implement and efficient. The tree traversal for an evaluation is comprised of a number of bit-mask operations and tests for coverage by the relevant child's basis function. This coverage operation can frequently be implemented much more efficiently than a basis function evaluation. For efficiency the non-Haar wavelet and scaling functions $\Gamma(x)$ are tabulated off line, and basis function evaluation becomes a table lookup after scaling and translating the parameter x. Haar bases are coded algorithmically because of their simplicity. Fig. 3 shows the traversal pseudo-code for a point-evaluation.

3.6.1 Hash Table

Another solution to the problem of storing internal zeros is to use a hash table representation of the array containing the non-zero coefficients. We use an open-addressing scheme keyed on the index of the coefficient[15]. This scheme is compact, requiring only storing the non-zero entries and the keys.

In an open addressing scheme the keys are stored in a table at an address given by some hash function $h(k, i)$ where k is the key, and i is a count of the number of probes required to find an empty slot. If the table is only lightly loaded most keys will hash to an empty slot on the first try. If a collision occurs, i is incremented and the process repeats. To do a lookup in the table the same sequence of hashes is examined until either the key is found at the position $h(k, i)$ in the table or an empty slot is found, at which time it is known that the key is not in the table, a negative probe.

In our application we expect most of our searches to have a negative result, indicating that the coefficient is zero. It is then important that negative searches be very quick. If the table is too full then negative probes will have to check a large number of entries and performance will be seriously affected.

Instead we allocate a table that is about twice as large as the number of non-zero coefficients. Since the expected number of probes required for a negative result is given by $1/(1-\alpha)$ where α is the load factor (ratio of full slots to table length), we find that with $\alpha = 0.5$ that a negative probe is expected to take 2 lookups. We feel that this is an acceptable trade-off.

4 Results

Our results fall into three categories: BRDF size reduction vs. error, fast low quality images, and the generation of high quality images. We will examine these in turn, after having described our test data.

4.1 Test Data

The BRDFs we used to test our method were generated using a virtual gonioreflectometer [26, 6]. This process gives us considerable control over the data generated. We can easily capture the same BRDF at different sampling resolutions and be assured that measurement errors are not an issue.

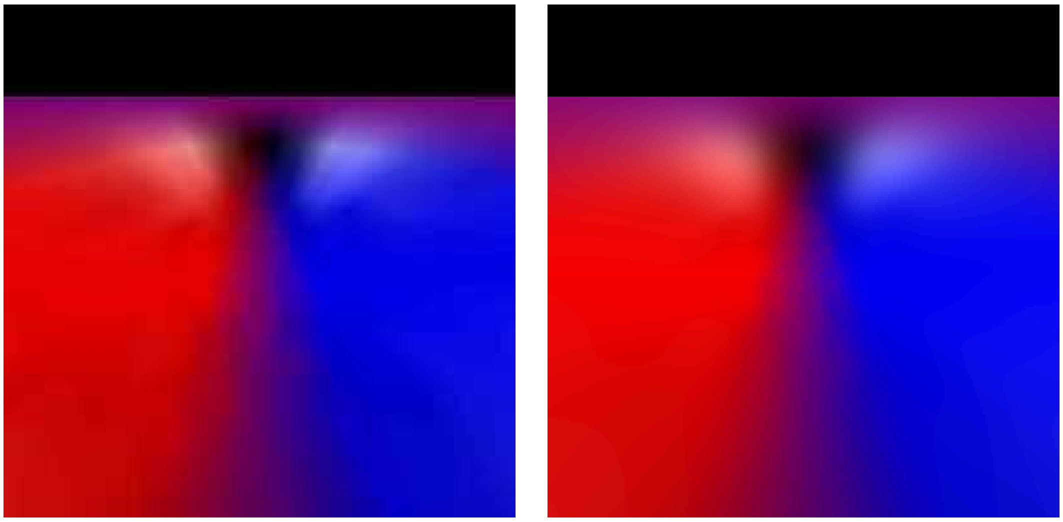
The BRDFs we examined in detail are:

- a sawtooth, whose micro-geometry is shown in Fig. 4;
- a brushed metal, made of a series of cylinders simulating scratches; and
- a Phong illumination model with an exponent of 30.

These were sampled at a resolution of sixteen samples for each of $\kappa_i, \lambda_i, \lambda_r$ and κ_r.

The measure of success of our BRDF representation is of compression vs. error, both in the representation of the BRDF, and in images shaded with the BRDF. Figure 5 shows compression vs. error rates for the sawtooth BRDF and various wavelet bases, where the error is the root-mean-square (RMS) of the difference between the complete BRDF and the thresholded BRDF. The important points to note from these plots is that the simple bases with narrow support perform quite competitively with the broader bases with more vanishing moments. In particular, the Daubechies basis with 10 vanishing moments only starts surpassing the simple linear spline[4] when the compression ratios drop to approximately one

[4] This is the spline 2,2 from page 277 of [5]



(a) (b)

Figure 2: Different interpolation methods for measured BRDFs. (a) quadrilinear interpolation; (b) tensor product cubic B-Spline; The micro-geometry, shown in Figure 4, is sampled at a resolution of 30 samples in ϕ_i and ϕ_r and 15 samples in θ_i and θ_r, in each of the red, green, and blue colour channels.

```
function Eval(BRDFTree brdf, vector d)
{
    result = 0;
    for b = 0 .. 15
        if (brdf->valuemask & (0x01>>b))
            j = (b,brdf->l,brdf->m[0..4]);
            result += brdf->value[b] * B_j(d);
        if (brdf->childMask & (0x01>>b))
            j = (b, brdf->level+1, brdf->m[0] * 2 + ((b>>0)&0x01),
                                   brdf->m[1] * 2 + ((b>>1)&0x01),
                                   brdf->m[2] * 2 + ((b>>2)&0x01),
                                   brdf->m[3] * 2 + ((b>>3)&0x01));
            // Check the support of B_j()
            if (InSupportOf_B_j(d))
                result += Eval(brdf->child[b], d);
    return result;
}
```

Figure 3: Reconstruction of a 4-variable Wavelet compressed function at point $\mathbf{d}$

percent of the original data size. The added cost of evaluating the Daubechies basis however (about 16 times the work — the Daubechies basis has about twice the coverage of the linear spline) makes this a gain of only marginal utility. Overall the linear spline bases seem to be adequate to reconstruct BRDF data to usable sizes with few visible artifacts (recall that the dual is used for the analysis, and that we are interested only in the width of the reconstruction wavelet).

Table 1 shows a comparison of running times and accuracy in the rendered image with thresholding. The image RMS error is computed from pixel values, scaled to the range (0..1], of the difference images, shown in Fig. 6.

4.2 Images

The suite of images in figure 6 shows a chair shaded with each of the BRDFs and at increasing levels of accuracy[5]. The corresponding difference images below them show the absolute difference from the chair rendered from the unthresholded BRDF. The degradation in quality is gradual, and the number of coefficients can be cut to a small fraction of the original data before serious artifacts occur. When a fast low-quality rendering is required a hihgly thresholded version of the data can be used; a more accurate version can be used if high-quality images are necessary.

5 Conclusion

We have presented a wavelet based compression method for BRDFs, along with an efficient data structure and point-reconstruction algorithm. We have shown that using the non-standard wavelet decomposition leads to considerable more efficient algorithms than using the standard decomposition. The compression method allows datasets to be reduced to 5% or 1% of their original sizes without serious artifacts. This means that using wavelet representations of BRDFs is practical for rendering scenes with large numbers of BRDFs.

In other work we have shown how to use our wavelet representation of the BRDF to generate reflected directions for Monte Carlo path tracing [17], and to filter the incident light at a point being shaded [16].

Other issues that must be addressed include the use of this BRDF representation with texture mapping and filtering techniques. The measured representations of BRDFs have in the past been too bulky to use as parts of texture maps, but our wavelet representation provides sufficient control over compactness that some pre-filtering techniques may become useful.

References

[1] James F. Blinn. Models of light reflection for computer synthesized pictures. volume 11, pages 192–198, July 1977.

[2] Brian Cabral, Nelson Max, and Rebecca Springmeyer. Bidirectional reflection functions from surface bump maps. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 273–281, July 1987.

[3] Michael Cohen, Donald P. Greenberg, Dave S. Immel, and Philip J. Brock. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications*, 6(3):26–35, March 1986.

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

[5] Ingrid Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, PA, 1992.

[6] Joel DeYoung, Paul Lalonde, and Alain Fournier. Acquiring and using realistic reflectance data in computer graphics images. In *Arkansas Computer Conference*, pages 77–82, 1996. Searcy, Arkansas: March 7–8, 1996.

[7] A. Fournier. Separating reflection functions for linear radiosity. In *Eurographics Rendering Workshop 1995*. Eurographics, June 1995.

[5] Chair data is courtesy of David Breen

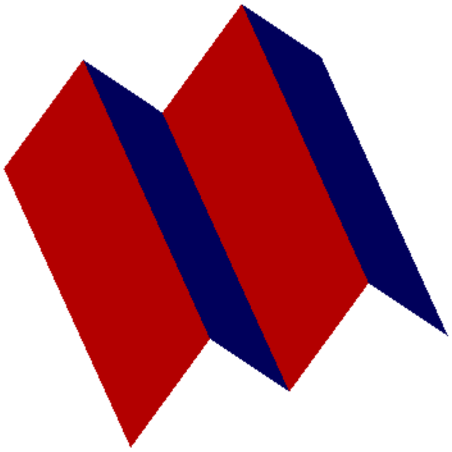
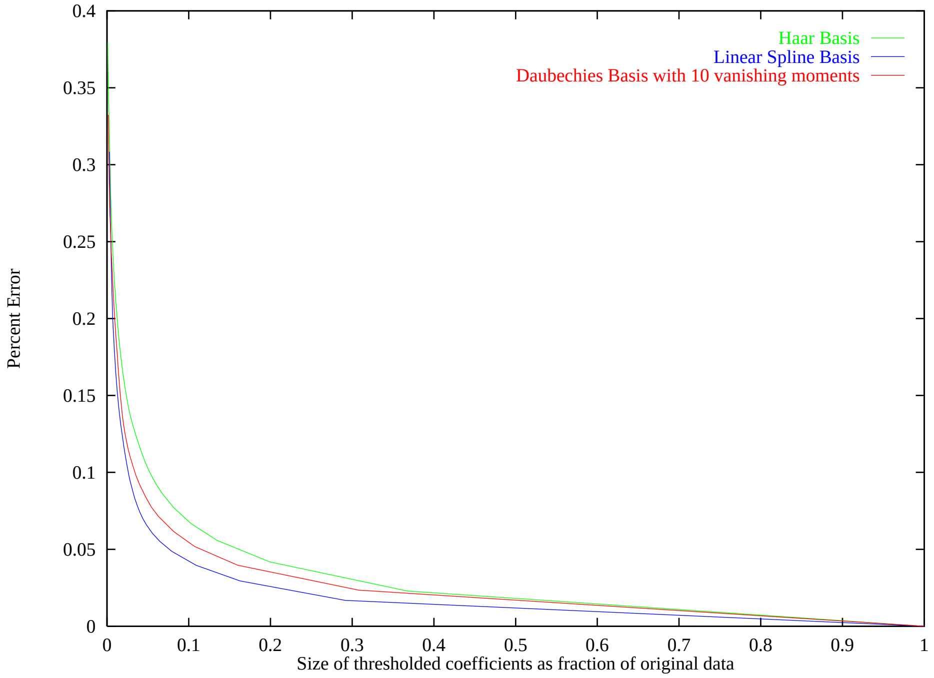


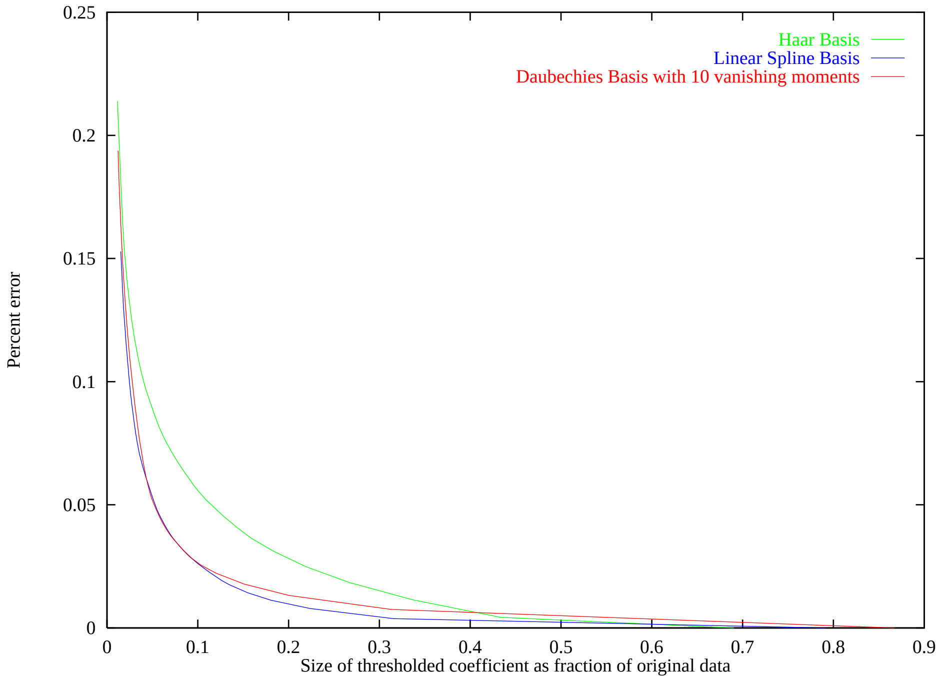
Figure 4: The sawtooth micro-geometry.

BRDF	Size	# Coef.	Time	Image RMS	BRDF RMS
Phong (n=50)	1.0	150933	50.9	0.0000	0.0000
	0.2	30183	36.4	0.0020	0.0244
	0.1	15051	30.1	0.0031	0.0549
	0.05	7542	22.7	0.0054	0.1049
	0.01	1503	12.5	0.0159	0.2334
	0.005	747	8.8	0.0236	0.2814
	0.001	147	3.5	0.0825	0.4331
	0.0001	12	1.5	0.1877	0.7486
Brushed	1.0	93911	30.0	0.0000	0.0000
	0.20	18779	16.2	0.0010	0.0437
	0.10	9388	13.2	0.0028	0.0944
	0.05	4692	10.9	0.0029	0.1612
	0.01	936	6.4	0.0152	0.2984
	0.005	467	5.5	0.0216	0.3535
	0.001	91	2.8	0.0362	0.5160
	0.0001	6	1.3	0.0776	0.8279
Sawtooth	1.0	127334	40.4	0.0000	0.0000
	0.2	25463	22.8	0.0075	0.0386
	0.1	12728	17.7	0.0149	0.0719
	0.05	6363	13.0	0.0202	0.1207
	0.01	1268	6.7	0.0352	0.3479
	0.005	632	5.2	0.0459	0.4639
	0.001	123	2.8	0.0762	0.6809
	0.0001	8	1.5	0.1486	0.9231

Table 1: Comparison of running times vs thresholding and accuracy. Size is given as the fraction of original non-zero coefficients; time is in milliseconds per pixel on a 200 mhz MIPS R4400 processor; RMS errors are calculated after normalization of data to the range (0..1]. The BRDF RMS is for the red channel. Green and blue behave similarly.

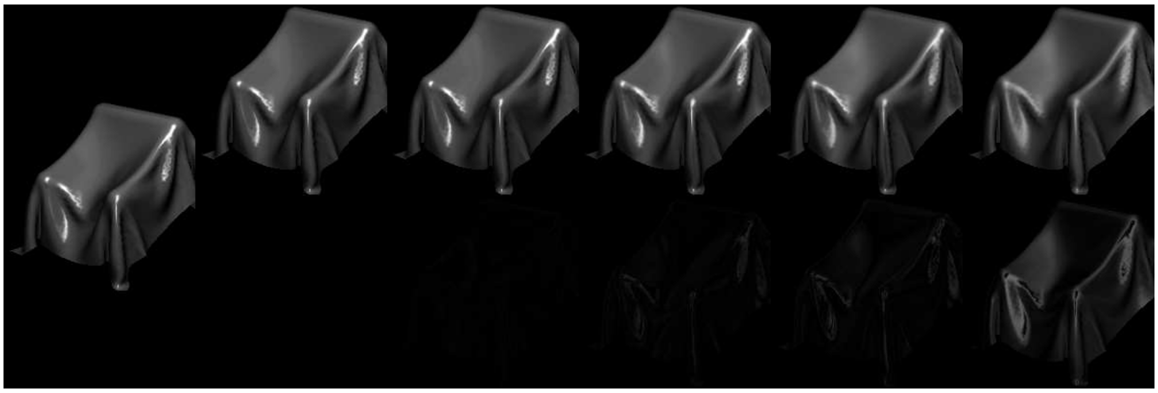


(a)

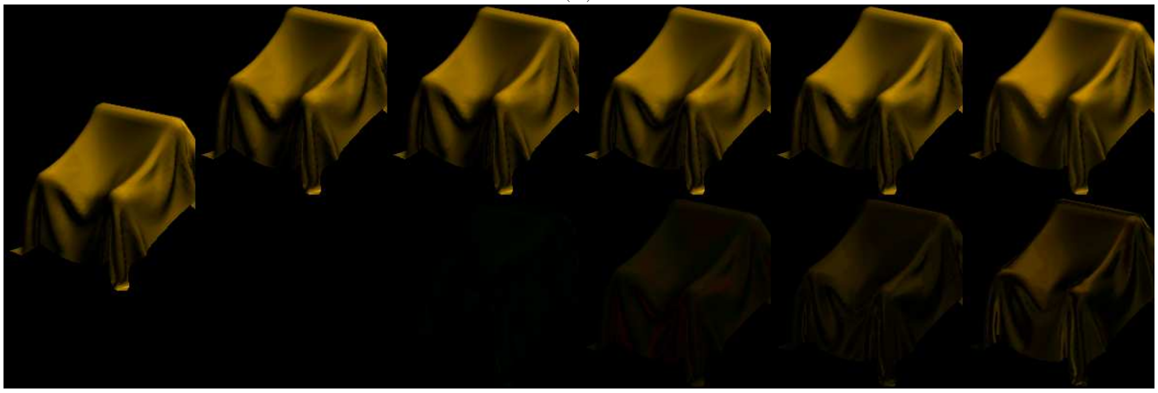


(b)

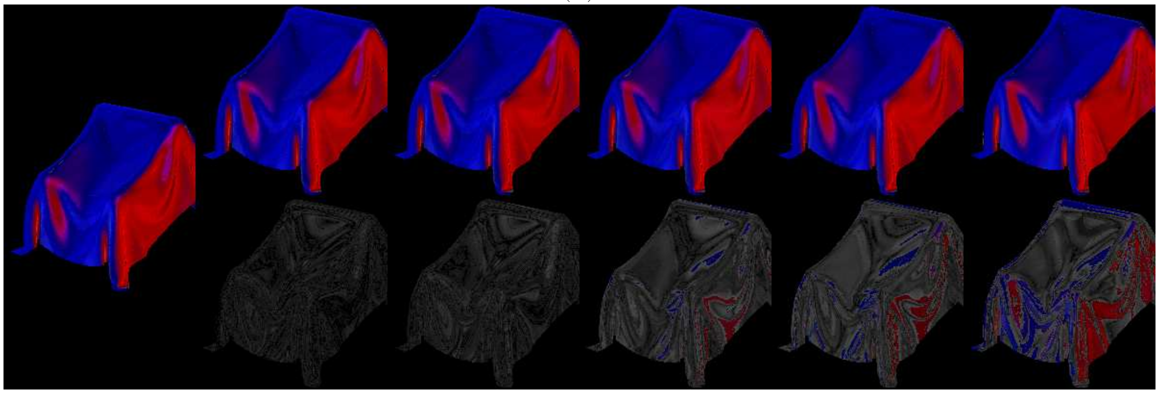
Figure 5: Compression vs. Percent total error: (a) for the sawtooth BRDF; (b) for a Phong illumination model with exponent 30.



(a)



(b)



(c)

Figure 6: A chair shaded with our wavelet compressed BRDFs, and the difference images. From left to right the images are generated with BRDFs thresholded to 10%, 5%, 1%, 0.5%, and 0.1% or their original sizes. The (a) is a phong shader with an exponent of 50,(b) a brushed metal whose micro-geometry is a set of cylinders, and (c) is a sawtooth, red on one side, blue on the other.

[8] Jay S. Gondek, Gary W. Meyer, and Jonathan G. Newman. Wavelength dependent reflectance functions. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 213–220. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.

[9] Steven J. Gortler, Peter Schroder, Michael F. Cohen, and Pat Hanrahan. Wavelet radiosity. In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 221–230, 1993.

[10] Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 165–174, August 1993.

[11] David S. Immel, Michael F. Cohen, and Donald P. Greenberg. A radiosity method for non-diffuse environments. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 133–142, August 1986.

[12] American National Standards Institute. ANSI standard nomenclature and definitions for illuminating engineering,. ANSI/IES RP-16-1986, Illuminating Engineering Society, 345 East 47th Street, New York, NY 10017, June 1986.

[13] James T. Kajiya. Anisotropic reflection models. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 15–21, July 1985.

[14] James T. Kajiya. The rendering equation. In David C. Evans and Russell J. Athay, editors, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20, pages 143–150, August 1986.

[15] D. E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison-Wesley, Reading, MA, 1973.

[16] Paul Lalonde and Alain Fournier. Filtered local shading the wavelet domain. In *Proceedings of the 8th Eurographics Workshop on Rendering*, St. Etienne, France, June 1997.

[17] Paul Lalonde and Alain Fournier. Generating reflected directions from brdf data. *Computer Graphics Forum*, volume 16, September 1997. Eurographics '97 Conference issue.

[18] Robert R. Lewis and Alain Fournier. Light-driven global illumination with a wavelet representation of light transport. In *Seventh Eurographics Workshop on Rendering*, Porto, Portugal, June 1996.

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

[20] Peter Schröder and Wim Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 161–172. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[21] Francois X. Sillion, James R. Arvo, Stephen H. Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 187–196, July 1991.

[22] Brian E. Smits and Gary Meyer. Newton'S colors: Simulating interference phenomena in realistic image synthesis. In *Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics*, pages 185–94, Rennes, France, June 1990.

[23] Eric J. Stollnitz, Tony D. DeRose, and David H. Salesin. Wavelets for computer graphics: A primer. *IEEE Computer Graphics and Applications*, 15(3), 1995.

[24] Gregory J. Ward. Measuring and modeling anisotropic reflection. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 265–272, July 1992.

[25] Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear. A ray tracing solution for diffuse interreflection. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 85–92, August 1988.

[26] Stephen H. Westin, James R. Arvo, and Kenneth E. Torrance. Predicting reflectance functions from complex surfaces. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 255–264, July 1992.

[27] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.