Eurographics Symposium on Rendering (2004)
A. Keller, H. W. Jensen (Editors)

All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation

Rui Wang, John Tran, and David Luebke

Department of Computer Science, University of Virginia

Abstract

This paper presents a technique, based on pre-computed light transport and separable BRDF approximation, for interactive rendering of non-diffuse objects under all-frequency environment illumination. Existing techniques using spherical harmonics to represent environment maps and transport functions are limited to low-frequency light transport effects. Non-linear wavelet lighting approximation is able to capture all-frequency illumination and shadows for geometry relighting, but interactive rendering is currently limited to diffuse objects. Our work extends the wavelet-based approach to relighting of non-diffuse objects. We factorize the BRDF using separable decomposition and keep only a few low-order approximation terms, each consisting of a 2D light map paired with a 2D view map. We then pre-compute light transport matrices corresponding to each BRDF light map, and compress the data with a non-linear wavelet approximation. We use modern graphics hardware to accelerate pre-computation. At run-time, a sparse light vector is multiplied by the sparse transport matrix at each vertex, and the results are further combined with texture lookups of the view direction into the BRDF view maps to produce view-dependent color. Using our technique, we demonstrate rendering of objects with several non-diffuse BRDFs under all-frequency, dynamic environment lighting at interactive rates.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics]: Color, shading, shadowing, and texture

1. Introduction

Interactive rendering of objects with realistic materials, complex illumination, and shadows continues to present a great challenge in computer graphics. Conventional image synthesis techniques such as Monte Carlo ray tracing [18, 38], photon mapping [17], and radiosity [7] simulate complex global illumination effects, but are too expensive for real-time rendering. Surface light fields [40, 6] allow for capturing realistic appearance and real-time rendering of physical objects; however, they do not allow for dynamic lighting. Our goal is to relight non-diffuse objects at interactive rates under dynamic, complex illumination and changing view.

Recently, Sloan et al. [36] introduced the pre-computed radiance transfer approach for shading models with low-frequency environment maps. They propose a compact representation of light transport functions using a spherical harmonic (SH) basis. Relighting then reduces to a simple inner product of the light vector, also represented in a SH basis, with pre-computed transport vectors at each vertex. View-dependent rendering of glossy surfaces uses a pre-computed transport matrix rather than a vector at each vertex. Although the technique is fast and compact, it is limited to low-frequency environment maps due to approximation with low-order (25D) spherical harmonics. Ng et al. [28] render all-frequency lighting and shadow effects at interactive frame rates by using a non-linear wavelet approximation of the environment map and transport matrix. However, for geometry relighting with changing view, their approach is limited to diffuse objects.

Our work extends Ng et al's geometry relighting technique to handle non-diffuse objects with complex BRDFs. We use a separable decomposition to approximate the BRDF with a few (K) low-order terms, each consisting of a 2D *light map* and 2D *view map*. At each vertex, we pre-compute K transport vectors corresponding to each of the K light maps. We compress the transport vectors using a non-linear

wavelet approximation as in [28] At run-time, when lighting changes, we sample the environment map and compute a sparse light vector using non-linear wavelet approximation. The sparse light vector is multiplied by the K transport vectors per vertex to produce K colors, which we call the *light-modulated K-tuple*. When the viewpoint changes, we use the view direction at each vertex to index into all K BRDF view maps, forming what we call the *view-modulated K-tuple*. Finally, the view-dependent color at each vertex is computed as the dot product of the view-modulated K-tuple with the light modulated K-tuple.

Kautz [19] and McCool [27] have proposed factorization techniques for interactive rendering with arbitrary BRDFs. We use a method similar to the singular value decomposition (SVD) method in [19], but exploit the fact that our BRDF matrix is real symmetric and have successfully experimented with using the *power iteration* algorithm for fast evaluation of low-order BRDF approximation. Our experiments show that the separable BRDF approximation is accurate enough for materials with moderate specular components (see Figure 4). Highly specular BRDF functions require more approximation terms, which reduces run-time performance.

Finally, we have accelerated pre-computation using modern graphics hardware, which decreases the pre-computation time by 50% compared with the CPU implementation.

2. Related Work

2.1. Pre-computation Techniques

The rendering equation [18] for distant direct illumination:

$$B(x,\omega_o) = \int_\Omega L(\omega_i)\, f_r(x,\omega_i \rightarrow \omega_o)\, V(x,\omega_i)\, (\omega_i \cdot \mathbf{n}(x))\, d\omega_i \tag{1}$$

describes light transport in the 6D space of incident direction ω_i, view direction ω_o and 2D surface location x. Due to its high dimensionality, the rendering equation is expensive to evaluate. Therefore, extensive research has examined pre-computation techniques for real-time rendering.

Environment maps were first introduced by Blinn and Newell [4] to approximate specular reflection of distant environments. Since then several approaches have been proposed to simulate diffuse and glossy reflection based on pre-convolved environment maps [14, 20, 31, 32]. However,by assuming Eq 1 is independent of surface location x and ignoring visibility V, they cannot handle shadowing or spatially varying BRDFs.

Image or texture relighting techniques pre-compute global illumination solutions for a set of lighting approximation functions, such as points [11], polynomials [26], steerable functions [30, 3] and compressed principal component basis [9]. These techniques can handle dynamic lighting changes and global illumination effects such as self-shadowing and interreflections. However, they assume that the view direction ω_o depends only on surface location x in Eq 1, thus requiring viewpoint to be fixed. Shadowing techniques have also been presented such as convolution textures [37], attenuation maps [1], and pre-computed visibility [16], but they do not allow real-time dynamic environment lighting and complex self-shadowing geometry.

Light fields [25, 13] record radiance samples as they pass through a pair of viewing planes. Surface light fields [40, 6] record exitant radiance in sampled directions over an object's surface. These techniques allow for view-dependent rendering of complex surface materials but are limited to static lighting since L is assumed to be fixed in Eq 1.

Recently, Sloan et al. introduced pre-computed radiance transfer (PRT), which captures the way an object shadows, scatters, and reflects light [36]. They pre-compute light transport functions (the product of f_r, V and $\omega \cdot \mathbf{n}$ in Eq 1) for each vertex and represent them as transport vectors in a low-order spherical harmonics (SH) basis. Relighting is then reduced to a simple inner product of the light vector, also represented in a SH basis, with the pre-computed transport vectors at each vertex. In the case of glossy surfaces, a transport matrix is pre-computed instead of a vector to support view-dependent rendering. This technique is further improved by [21, 24, 35] to allow for real-time rendering of arbitrary glossy surfaces.

Spherical harmonics have previously been applied by [34, 5, 39] as a compact representation for BRDF or global illumination solutions. However, as Ng et al. point out [28], a low-order SH basis can only approximate very low-frequency environment maps, and therefore cannot reproduce all-frequency lighting and shadowing effects. Instead, they propose using non-linear wavelet approximation to represent the environment map and light transport matrix, and achieve all-frequency illumination and shadows at interactive frame rates. They demonstrate the technique primarily for image relighting, and geometry relighting is limited to diffuse objects only. In their recent work [29] Ng et al. developed triple product integrals for all-frequency relighting. This new technique is accurate but rendering requires a few seconds. Our paper extends the wavelet-based approach of Ng et al. to all-frequency relighting of non-diffuse objects at interactive frame rates.

2.2. BRDF Factorization

BRDF factorization has been applied for interactive shading of realistic materials and important sampling of BRDFs [23] etc. Existing techniques for arbitrary BRDF factorization include Separable BRDF Decomposition [19] and Homomorphic Factorization [27]. They store the separated as BRDF 2D texture maps, then use graphics hardware to index these maps and render objects in real-time. Non-Negative Matrix Factorization (NMF) [8] can also be applied to decompose BRDFs, which generates strictly positive coefficients as the Homomorphic Factorization method.

In [10, 12] researchers examine separability and other properties of tabulated BRDFs. In [33, 19] researcher show that proper parametrization (such as the *half-angle* parametrization) of a BRDF can improve its separability remarkably.

The current mathematical framework of our technique (see Section 3.1) requires separable approximation of BRDFs parameterized by incident and view directions. We show that with a few low-order terms, our approximation is visually good enough for BRDFs with moderate specular components. The separable decomposition approximates a multivariate BRDF function f_r as a sum of products of functions g_k and h_k of lower dimensionality:

$$f_r(\omega_i, \omega_o) \approx \sum_{k=1}^{K} g_k(\omega_i)\, h_k(\omega_o) \tag{2}$$

where K is number of approximation terms. We call g_k the BRDF *light map* since it is a 2D texture map indexed with the incident direction ω_i. Similarly h_k is called *view map* since it is a 2D texture map indexed with the view direction ω_o. To decompose the BRDF, we use a similar method as the SVD method in [19], but exploit the knowledge that our BRDF matrix is real symmetric. As a result, the BRDF is approximated by a few low-order terms, each consisting of a 2D light map paired with a 2D view map.

3. Algorithms and Implementation

In this section, we describe algorithms for pre-computation and relighting using non-linear wavelet and separable BRDF approximations. We also discuss ways to accelerate pre-computation using modern graphics hardware.

3.1. Overview

We make the assumption that illumination is distant and direct, and we do not consider inter-reflected illumination. For convenience, we re-write Eq 1 here:

$$B(x, \omega_o) = \int_\Omega L(\omega_i)\, f_r(x, \omega_i \rightarrow \omega_o)\, V(x, \omega_i)\, (\omega_i \cdot \mathbf{n}(x))\, d\omega_i$$

where x is the surface location, B is the reflected light, L is the lighting, ω_i is incident direction, ω_o is view direction, f_r is the BRDF, V is visibility, and $\mathbf{n}$ is the surface normal at x. The integration domain is over the upper hemisphere with respect to $\mathbf{n}$. We assume L represents distant illumination (environment map). For simplicity, we consider the BRDF to be uniform across the object surface, but it is straightforward to handle spatially varying BRDFs modulated by a texture map or a linear combination of basic BRDFs, since our pre-computation is per vertex. Substituting Eq 2 (separable BRDF approximation) into Eq 1 gives:

$$\begin{aligned} & B(x, \omega_o) \\ &= \int_\Omega L(\omega_i) \left(\sum_{k=1}^{K} g_k(\omega_i)\, h_k(\omega_o) \right) V(x, \omega_i)\, (\omega_i \cdot \mathbf{n}(x))\, d\omega_i \\ &= \sum_{k=1}^{K} \left(h_k(\omega_o) \int_\Omega L(\omega_i)\, g_k(\omega_i)\, V(x, \omega_i)\, (\omega_i \cdot \mathbf{n}(x))\, d\omega_i \right) \end{aligned}$$

We define a combined transport function for the k-th BRDF term as:

$$T_k(x, \omega_i) = g_k(\omega_i)\, V(x, \omega_i)\, (\omega_i \cdot \mathbf{n}(x)) \tag{3}$$

which allows us to express reflected light as:

$$B(x, \omega_o) = \sum_{k=1}^{K} \left(h_k(\omega_o) \int_\Omega L(\omega_i)\, T_k(x, \omega_i)\, d\omega_i \right) \tag{4}$$

The integrand in the above equation is no longer dependent on ω_o. Using numerical cubature (with appropriate normalizing weights for integral transformation), we can write:

$$\int_\Omega L(\omega_i)\, T_k(x, \omega_i)\, d\omega_i = \sum_j T_k(x, \omega_j)\, L(\omega_j) \tag{5}$$

Finally, combining Eq 5 with Eq 4 we have:

$$B(x, \omega_o) = \sum_{k=1}^{K} \left(h_k(\omega_o) \sum_j T_k(x, \omega_j)\, L(\omega_j) \right)$$

For a given surface location x and its view direction ω_o, we can write the equation above in matrix notation:

$$B_x = \mathbf{h}_x \mathbf{T}_x \mathbf{L} \tag{6}$$

Here B_x is the view-dependent color of x, $\mathbf{L}$ is the light vector, $\mathbf{h}_x$ is called the *view-modulated K-tuple*, which is a K-vector evaluated by texture lookups of ω_o into all K BRDF view maps. $\mathbf{T}_x$ is the transport matrix, the rows of which are K transport vectors corresponding to the BRDF light maps. We define $\mathbf{g}_x = \mathbf{T}_x \mathbf{L}$, which is a K-vector called the *light-modulated K-tuple*. Now B_x is simply a dot product of $\mathbf{h}_x$ with $\mathbf{g}_x$. Clearly, changing the light requires re-evaluation of $\mathbf{g}_x$, but changing the view only requires updating $\mathbf{h}_x$. We discuss the runtime cost for each case in Section 4.2.

Eq 6 is similar to the matrix notation in [28]. The difference is that here the transport vectors are modulated by the BRDF light maps, and rendering is modulated by the BRDF view maps. As pointed out by Ng et al in [28], the illumination can be parameterized over any domain such as a sphere, a cubical environment map, or local lighting from a ceiling or window. Also, Eq 6 holds when L and row vectors of $\mathbf{T}_x$ are expressed in any orthonormal basis, such as spherical harmonics or wavelets.

3.2. Separable BRDF Approximation

Factorization We use the SVD technique to factorize BRDFs [19]. We sample the BRDF and construct a BRDF matrix $\mathbf{M}$, the columns and rows of which are sampled incident directions and view directions respectively. We process the R, G, and B color channels separately. Applying SVD on $\mathbf{M}$ gives $\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, which can be re-written as:

$$\mathbf{M} = \sum_{k=1}^{K} \sigma_k\, \mathbf{u}_k\, \mathbf{v}_k^T$$

where singular values σ_k are diagonal elements of $\mathbf{S}$, and $\mathbf{u}_k$, $\mathbf{v}_k$ are column vectors of $\mathbf{U}$ and $\mathbf{V}$. $\sqrt{\sigma_k}\mathbf{u}_k$ and $\sqrt{\sigma_k}\mathbf{v}_k$ correspond to BRDF light maps and view maps respectively. Since $\mathbf{u}_k$ and $\mathbf{v}_k$ are sets of orthonormal vectors, truncating the sum with the largest K singular values results in an optimal approximation of $\mathbf{M}$ in Frobenius norm.

In fact, due to Helmholtz reciprocity of BRDFs and the fact that we apply equal sampling resolution on both incident and view directions, the matrix $\mathbf{M}$ in our case is real symmetric. According to the Spectral Theorem, $\mathbf{M}$ is diagonalizable such that $\mathbf{M} = \mathbf{Q}\mathbf{D}\mathbf{Q}^T$. The diagonal elements of $\mathbf{D}$ are eigenvalues of $\mathbf{M}$, and the column vectors of $\mathbf{Q}$ are the corresponding normalized eigenvectors. They relate to the SVD of $\mathbf{M}$ by: $\mathbf{U} = \mathbf{V} = \mathbf{Q}$, $\mathbf{S} = \mathbf{D}^2$. Given this, we can apply the *power iteration* algorithm [15] on $\mathbf{M}$ for finding dominant matrix eigenvalues, which is fast and accurate for low-order approximation. Pseudocode for the algorithm is given in the Appendix. Notice that since $\mathbf{U} = \mathbf{V} = \mathbf{Q}$, the computed light map and view map will differ at most by a sign where the corresponding eigenvalue is negative.

In practice, a single term approximation ($K = 1$) can often reproduce a reasonable degree of specularity, and can be efficiently computed with the Normalized Decomposition [19]. However, for highly specular BRDFs, more approximation terms are necessary because the singular values tend to become equally large. We have experimented with $K = 4$ for all our BRDFs and found this to be visually good enough for materials with moderate specular components. In Section 4.3, we discuss the BRDF approximation error from varying K.

Parametrization The current mathematical framework in Section 3.1 requires the incident-view parametrization of BRDFs. Highly specular BRDFs are better handled by other (e.g. the *half-angle*) parametrization, which we cannot use currently. Light maps and view maps from the separated BRDF are stored as 2D textures, which are parameterized by the orthographic projection of a direction $\mathbf{d}$ from the upper hemisphere onto the unit disk:

$$\mathrm{x} = (\mathbf{d}\cdot\mathbf{s}+1)/2 \qquad \mathrm{y} = (\mathbf{d}\cdot\mathbf{t}+1)/2 \tag{7}$$

where x and y are texture coordinates, and $\mathbf{s}$ and $\mathbf{t}$ are tangent vectors of the local coordinate system. Unless the object is parameterized, the tangent vectors $\mathbf{s}$ and $\mathbf{t}$ must be evaluated from the normal $\mathbf{n}$. We use *stereographic projection* of the unit sphere to parameterize normal space and derive the formula:

$$\begin{aligned}\mathbf{s} = {} & [\ (\mathtt{n_y}+1)^2 + \mathtt{n_z}^2 - \mathtt{n_x}^2, \\ & -2(\mathtt{n_y}+1)\mathtt{n_x}, \quad -2\mathtt{n_x}\mathtt{n_z}\ ].\mathtt{normalize} \\ \mathbf{t} = {} & \mathbf{n}\times\mathbf{s}\end{aligned}$$

where $\mathbf{n}$=[$\mathtt{n_x}$, $\mathtt{n_y}$, $\mathtt{n_z}$] is an arbitrary normal. Compared with the common longitude-latitude parametrization, which leads to two singularities generated for $\mathbf{s}$ and $\mathbf{t}$ when $\mathbf{n}$=[0, 1, 0] or [0, −1, 0], our method produces only one singularity at $\mathbf{n}$=[0, −1, 0]. Although we typically use the orthographic parametrization, we found that for some BRDFs such as the Ashikhmin-Shirley [2] anisotropic BRDF, the polar coordinates parametrization using the following formula gives significantly better single term approximation:

$$\mathrm{x} = \tan^{-1}(\mathbf{d}\cdot\mathbf{s}, \mathbf{d}\cdot\mathbf{t}) \qquad \mathrm{y} = \sqrt{1-\mathbf{d}\cdot\mathbf{n}} \tag{8}$$

3.3. Pre-computation

We pre-compute a light transport matrix $\mathbf{T}_x$ at each vertex. Distant lighting is parameterized using $6\times64\times64$ cubical environment maps. Incident and view directions are both expressed in the same global coordinate system as lighting. First, we rasterize a high resolution ($6\times256\times256$) visibility cubemap at each vertex. We then use the direction vector of each cubemap pixel to index into the BRDF light maps, and evaluate the transport function as the product of visibility, light map values and the cubemap pixel weight (i.e. the integral transformation factor). The result is then down-sampled to $6\times64\times64$, which is the environment map resolution.

Wavelet Transform We project each cubemap face (64×64) onto the 2D Haar wavelet basis. In the case of a K-term BRDF approximation, this requires $6\times K$ wavele transforms. For non-linear approximation, we gather the wavelet coefficients on all 6 cubemap faces and apply a linear-time selection algorithm to gather and store the first N_l wavelet coefficients according to their magnitudes. Because the wavelet bases are orthonormal, keeping the coefficients with the largest magnitudes is an optimal approximation for a L2-norm. The K transport vectors at each vertex are processed separately. We have experimented with $N_l = 96$ (about 0.4% of uncompressed data) and achieved high rendering fidelity. In Section 4.3 we show the wavelet approximation error visually by varying N_l.

Quantization and Storage To further reduce our data size and improve rendering speed, we uniformly quantize the wavelet coefficients. The three color channels are quantized to signed 11, 11 and 10 bits each and are packed into a 4-byte field. The indices for our sparse transport vectors are stored separately as 16-bit unsigned integer. On average the quantization process reduces data size to $\frac{1}{4}$ of un-quantized data. The rendering speed is more than doubled due to reduced data size and integer operations in place of floating point operations.

Hardware Acceleration The pre-computation algorithm is well suited for acceleration on graphics hardware. To do this, we render the visibility map into an OpenGL pbuffer, and bind it as a texture for a shader that computes the transfer function. This transfer function is rendered into a floating point render target, and is bound again to do the downsampling on the card. We then only read data off of the card in 64×64 blocks of memory. To reduce memory bandwidth cost for data transfer between the CPU and GPU, we store

the visibility model on the card. The cube map direction normals and BRDF approximation terms are stored as floating point textures. Because the GPU lacks floating point interpolation support, the fragment shaders must also perform a bilinear interpolation when indexing into the textures.

3.4. Rendering

At run-time, we sample the environment map dynamically on a $6 \times 64 \times 64$ cubemap, perform a fast 2D Haar wavelet transformation and then uniformly quantize the wavelet coefficients to signed 16-bits per color channel. Quantized coefficients with absolute value below a certain threshold (1024 in our experiments) are thrown away. This results in keeping about $200 \sim 600$, or $0.8\% \sim 2.5\%$ of the lighting coefficients, depending on the high frequency information present in the lighting. Since we keep much more lighting coefficients than the transport function coefficients at each vertex, the error of rendering due to non-linear approximation of the lighting is less notable. Therefore we use unweighted selection of the lighting coefficients [28].

To relight the scene, we perform a multiplication of the sparse transport matrix with the sparse light vector, and produce the light-modulated K-tuple for each vertex. The view direction at each vertex is used to index into BRDF view maps and produce the view-modulated K-tuple. Vertex color is computed as the dot product of the two tuples. Notice that the light-modulated K-tuple is re-computed only when lighting changes, which involves more expensive computation than changing view.

4. Results and Discussion

4.1. Separable BRDF Approximation

We present results using the following BRDFs:

- A red plastic and light brown plastic BRDF
- Ashikhmin-Shirley (AS) anisotropic BRDF [2] with parameters $k_d = 0.5$, $k_s = 0.75$, $n_u = 10$, $n_v = 100$
- A clay BRDF and a dark skin BRDF, using the Lafortune model [22]

To construct the BRDF matrix, we sample both incident and view directions on a unit disk embedded in a 2D texture map of 48×48 resolution. The memory required to store the tabulated BRDF is approximately 37.5 MB. Sampling at a higher resolution is possible, but requires more memory and computation time, while providing little improvement in accuracy (see Section 4.3). Notice that the cosine term in the transport function (Eq 6) could be built into the BRDF, however, for some BRDFs it can affect the separability. Figure 4 shows the approximation results for four BRDFs. In each image, we show the first 8 BRDF terms, produced by the algorithm in Section 3.2, along with a rendering of the bunny model using only the first 4 terms. The bunny is illuminated by various environment maps. These examples demonstrate that a 4-term approximation is visually adequate for materials with moderate specular components.

Since matrix SVD usually produces some negative coefficients, we occasionally notice undesired negative residue colors (such as the noticeable blue colors around the bottom of Lucy's skirt in Figure 7).

4.2. Performance

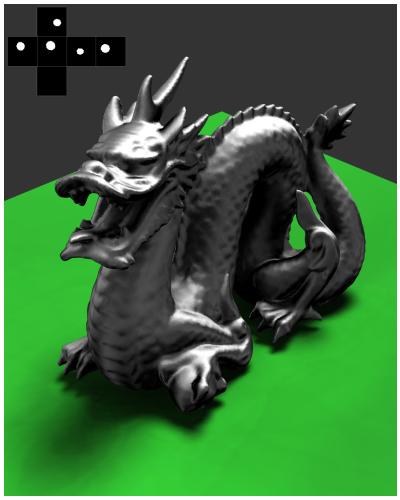
Our test results were acquired on an Intel Pentium 4 2.6 GHz computer with 1 GB memory and an Nvidia GeForce 5900 graphics card. We compile our programs using the Intel Compiler version 7.1, and we use the OpenGL vertex buffer objects (VBO) extension to reduce memory bandwidth cost for data transfer between CPU and GPU.

Pre-computation Table 1 shows the pre-computation times and storage requirements for a number of models used in our experiments. All models are pre-computed with $K = 4$ BRDF approximation terms. For comparison, we also list the pre-computation time and storage requirements for single term $K = 1$ approximation. To demonstrate the ability of our technique to handle all-frequency shadows, we render a diffuse ground quad under each model; the performance numbers include pre-computation time for the ground.

For visibility sampling, we use a simplified scene model to reduce rasterization time. This approximation has no visible effect on any of our test scenes. Our non-linear wavelet approximation keeps up to $N_l = 96$ largest coefficients for each transport vector. Coefficients that are quantized to zero are thrown away. Since we evaluate a transport matrix, consisting of $K = 4$ transport vectors per vertex, the total storage requirement is roughly 200 MB for a 100,000-vertex model. The size is roughly $\frac{1}{4}$ of un-quantized data. By using graphics hardware acceleration, we are able to reduce pre-computation time by 50% compared to a CPU implementation, which uses the graphics hardware only for visibility sampling.

Rendering Table 2 lists our rendering frame rates. For view change, we achieve interactive rates for all our models. For lighting change, we achieve interactive rates for small models ($\sim$ 50K vertices) and near interactive rates for bigger models ($\sim$ 150K vertices).

Figure 6 shows the ability of our technique to handle all-frequency illumination effects. The Buddha model is illuminated by the *St. Peter's Basilica* environment map. Notice the clear definition of shadows on the ground. The Lucy model is illuminated by a small area light. This high-frequency lighting creates a sharp contrast between Lucy's bright front face and dark back face. Also notice the dark shadow on the ground. Techniques based on low-order spherical harmonics representation cannot reproduce such high-frequency illumination effects. Figures 1 and 7 show several models rendered with different lighting and BRDFs.



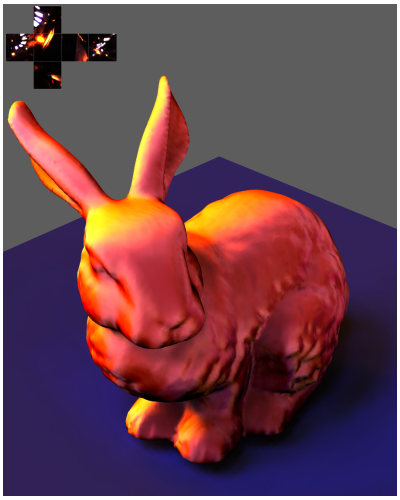


Figure 1: *In the left the Dragon model is rendered with the Ashikhmin-Shirley anisotropic BRDF and illuminated by the* Uffizi Gallery *environment map; in the right the Bunny model is rendered with a red plastic BRDF and illuminated by the* Grace Cathedral *environment map.*

	Bunny	Buddha	Dragon	Lucy
Vertices	36 K	56 K	100 K	180 K
Ground	22 K	31 K	63 K	23 K
Simp. Vis.	11 K	20 K	32 K	17 K
1-term P.T.	21 min	35 min	70 min	80 min
4-term P.T.	70 min	101 min	194 min	233 min
1-term Size	30 MB	44 MB	81 MB	100 MB
4-term Size	84 MB	127 MB	230 MB	370 MB

Table 1: *Listed here are: model size (number of vertices), size of the diffuse ground, size of the simplified scene model for visibility sampling, 1-term and 4-term pre-computation time, and corresponding storage size.*

	Bunny	Buddha	Dragon	Lucy
1-term View	90 fps	46 fps	24 fps	18 fps
4-term View	63 fps	35 fps	18 fps	12 fps
1-term Light	20 fps	14 fps	7.8 fps	6.2 fps
4-term Light	7.2 fps	5.0 fps	2.7 fps	1.7 fps

Table 2: *Rendering frame rates for view change and lighting change, with 1-term and 4-term BRDF approximation.*

4.3. Accuracy

BRDF Approximation To analyze the accuracy of our BRDF approximation, we choose 80,000 randomly distributed samples over the incident and view direction, evaluate the approximated BRDF value, and then compute the root mean squared (RMS) error of the samples with the analytic BRDF, weighted by the cosine term $(\omega_i \cdot \mathbf{n})$ of the incident direction. In Figure 4, we plot the cosine-weighted RMS error of each BRDF approximation as function of the number of approximation terms; each curve represents a different light-view map resolution. The non-zero asymptotic error is due to bilinear interpolation of finite resolution texture maps. Notice that in all the examples, the error line of 32×32 texture resolution is very close to that of 64×64 texture resolution. Sampling at a higher resolution requires more memory and computation time, while providing little improvement in accuracy. Therefore we choose 48×48 texture resolution, which is both accurate and fast to compute.

It is well-known that RMS error of the BRDF function does not directly relate to the error in rendering. In Figure 2 we show rendering of the Buddha model with different number of approximation terms (K) of the Ashikhmin-Shirley anisotropic BRDF. The model is illuminated by a single distant light. The rightmost image is rendered with the analytic BRDF for reference. Comparing fidelity of the specular highlights, we see that a 4-term approximation is visually accurate. Increasing the number of terms used in pre-computation requires more file storage size and reduces rendering speed. Therefore we choose $K = 4$, which is visually accurate and results in acceptable resource requirement.

In case where accuracy is not the primary concern, we found that a single term approximation can often reproduce a reasonable amount specular components. Figure 5 compares the Lucy model rendered with a diffuse gray BRDF, a 1-term approximation and 4-term approximation of the Ashikhmin-Shirley anisotropic BRDF. Notice the specular highlights that the single term approximation exhibits compared to the diffuse BRDF.

Non-linear Wavelet Approximation Figure 3 shows several renderings of the Bunny model with a high-frequency shadow on the ground as we vary the number of wavelet approximation terms (N_l) in the transport function. We see that $N_l = 96$ gives visually good result. Increasing N_l gives higher fidelity of the high-frequency shadows but also requires more computation time and storage size. Since we

keep much more lighting wavelet coefficients than the transport function coefficients at each vertex, the error in rendering due to the lighting approximation is less notable. [28] is a good reference for accuracy analysis of non-linear wavelet approximation of lighting.

5. Conclusion and Future Work

We have shown that using non-linear wavelet approximation and separable BRDF approximation enables rendering of non-diffuse objects under all-frequency illumination at interactive rates.

We are currently working on accelerating rendering using programmable graphics hardware. Our current implementation uses a vertex shader to compute the texture coordinates based on the view direction and the tangent vectors (see Eq 7), and a fragment shader performs the dot product of the view-modulated K-tuple with the light-modulated K-tuple to produce a per pixel color, using floating point texture maps to represent the BRDF terms. Although our prototype GPU implementation is interactive, it can be outperformed by a carefully tuned CPU implementation, using the Intel SSE2 instruction set on a high-end PC. For pre-computation, we plan to experiment with principal component analysis methods such as [35, 24] to further reduce the storage size by exploiting data coherence among vertices. As an extension to our current pre-computation algorithm on graphics hardware, we also plan to compute the transfer function of multiple vertices at the same time. By careful load-balancing, we hope to offload an equal amount of work to the GPU as the CPU, allowing them to operate at full speed asynchronously.

Acknowledgements The models used are courtesy of the Stanford 3D Scanning Repository. We would like to thank Nolan Goodnight for advice on our work.

References

[1] M. Agrawala, R. Ramamoorthi, A. Heirich, and L. Moll. Efficient image-based methods for rendering soft shadows. In *Proc. SIGGRAPH'00*, pages 375–384, 2000. 2

[2] M. Ashikhmin and P. Shirley. An anisotropic phong brdf model. *J. Graph. Tools*, 5(2):25–32, 2000. 4, 5

[3] M. Ashikhmin and P. Shirley. Steerable illumination textures. *ACM Trans. Graph.*, 21(1):1–19, 2002. 2

[4] J. F. Blinn and M. E. Newell. Texture and reflection in computer generated images. *Commun. ACM*, 19(10):542–547, 1976. 2

[5] B. Cabral, N. Max, and R. Springmeyer. Bidirectional reflection functions from surface bump maps. In *Proc. SIGGRAPH'87*, pages 273–281, 1987. 2

[6] W.-C. Chen, J.-Y. Bouguet, M. H. Chu, and R. Grzeszczuk. Light field mapping: efficient representation and hardware rendering of surface light fields. In *Proc. SIGGRAPH'02*, pages 447–456, 2002. 1, 2

[7] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, 1993. 1

[8] L. DD and S. HS. Learning the parts of objects by non-negative matrix factorization. *Nature 401*, pages 788–791, 1999. 2

[9] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the reflectance field of a human face. In *Proc. SIGGRAPH'00*, pages 145–156, 2000. 2

[10] J. DeYoung and A. Fournier. Properties of tabulated bidirectional reflectance distribution functions. In *Proc. Graphics Interface '97*, pages 47–55, 1997. 3

[11] J. O. Dorsey, F. X. Sillion, and D. P. Greenberg. Design and simulation of opera lighting and projection effects. In *Proc. SIGGRAPH'91*, pages 41–50, 1991. 2

[12] A. Fournier. Separating reflection functions for linear radiosity. In *6th Eurographics Rendering Workshop*, pages 383–392, 1995. 3

[13] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proc. SIGGRAPH'96*, pages 43–54, 1996. 2

[14] N. Greene. Environment mapping and other applications of world projections. *IEEE Comput. Graph. Appl.*, 6(11):21–29, 1986. 2

[15] M. T. Heath. *SCIENTIFIC COMPUTING: An Introductory Survey*. McGraw-Hill, 2002. 4

[16] W. Heidrich, K. Daubert, J. Kautz, and H.-P. Seidel. Illuminating Micro Geometry Based on Precomputed Visibility. In *Proc. SIGGRAPH'00*, pages 455–464, July 2000. 2

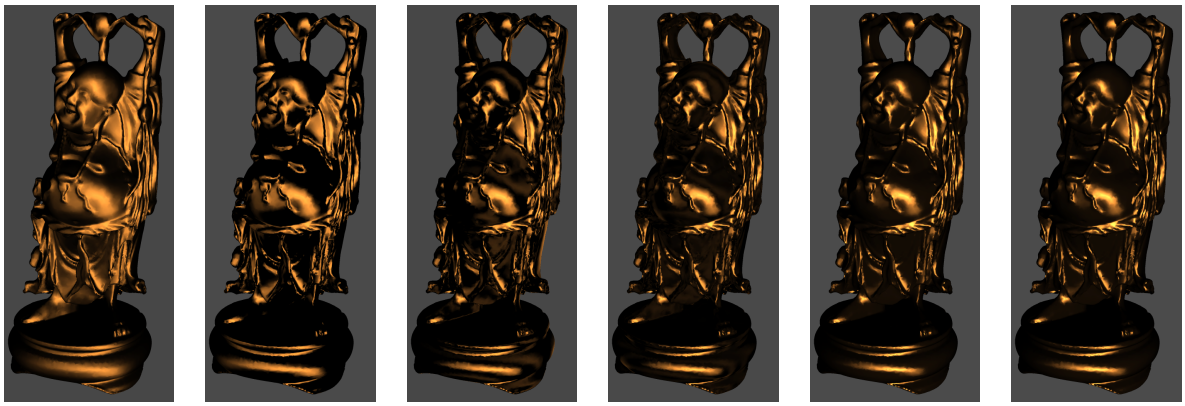
[17] H. W. Jensen. Global Illumination Using Photon Maps. In *7th Eurographics Rendering Workshop*, pages 21–30, 1996. 1

[18] J. T. Kajiya. The rendering equation. In *Proc. SIGGRAPH'86*, pages 143–150, 1986. 1, 2

[19] J. Kautz and M. D. McCool. Interactive Rendering with Arbitrary BRDFs using Separable Approximations. In *10th Eurographics Rendering Workshop*, pages 281–292, 1999. 2, 3, 4

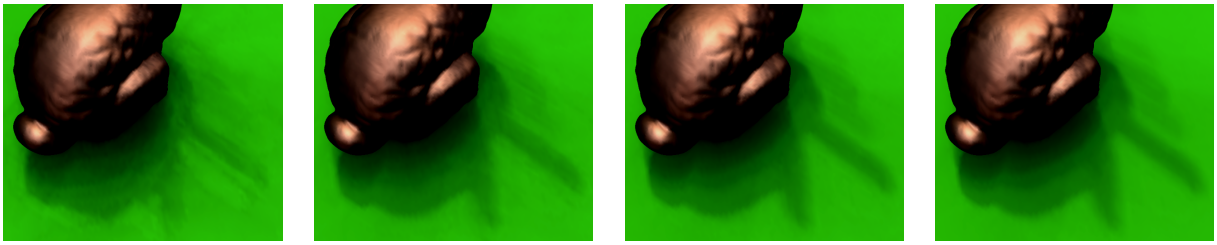
[20] J. Kautz and M. D. McCool. Approximation of glossy reflection with prefiltered environment maps. In *Graphics Interface 2000*, pages 119–126, 2000. 2

[21] J. Kautz, P.-P. Sloan, and J. Snyder. Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *13th Eurographics Rendering Workshop*, pages 291–296, 2002. 2



K =**1** K =**2** K =**4** K =**16** K =**64** *Reference*

Figure 2: *The Buddha model rendered with the Ashikhmin-Shirley anisotropic BRDF, illuminated by a single distant lighting. We compare fidelity of the specular highlight by increasing the number of BRDF approximation terms K. The rightmost shows rendering with the analytic BRDF.*



$N_l = 64\,(0.26\%)$ $N_l = 96\,(0.4\%)$ $N_l = 160\,(0.5\%)$ $N_l = 256\,(1\%)$

Figure 3: *The Bunny model casting shadows on the ground. We compare fidelity of the shadows by increasing the number of wavelet approximation terms N_l in the pre-computed transport vectors. Percentage numbers are given in parentheses.*

[22] E. P. Lafortune, S.-C. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance functions. In *Proc. SIGGRAPH'97*, volume 31, pages 117–126, 1997. 5

[23] J. Lawrence, S. Rusinkiewicz, and R. Ramamoorthi. Efficient brdf importance sampling using a factored representation. *To Appear at ACM Trans. Graph.*, 2004. 2

[24] J. Lehtinen and J. Kautz. Matrix radiance transfer. In *Proc. Symposium on Interactive 3D graphics 2003*, pages 59–64, 2003. 2, 7

[25] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. SIGGRAPH'96*, pages 31–42, 1996. 2

[26] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *Proc. SIGGRAPH'01*, pages 519–528, 2001. 2

[27] M. D. McCool, J. Ang, and A. Ahmad. Homomorphic factorization of brdfs for high-performance rendering. In *Proc. SIGGRAPH'01*, pages 171–178, 2001. 2

[28] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph.*, 22(3):376–381, 2003. 1, 2, 3, 5, 7

[29] R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product wavelet integrals for all-frequency relighting. *To Appear at ACM Trans. Graph.*, 2004. 2

[30] J. S. Nimeroff, E. Simoncelli, and J. Dorsey. Efficient Re-rendering of Naturally Illuminated Environments. In *5th Eurographics Rendering Workshop*, pages 359–373, 1994. 2

[31] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proc. SIGGRAPH'01*, pages 497–500, 2001. 2

[32] R. Ramamoorthi and P. Hanrahan. Frequency space environment map rendering. In *Proc. SIGGRAPH'02*, pages 517–526, 2002. 2

[33] S. M. Rusinkiewicz. A new change of variables for

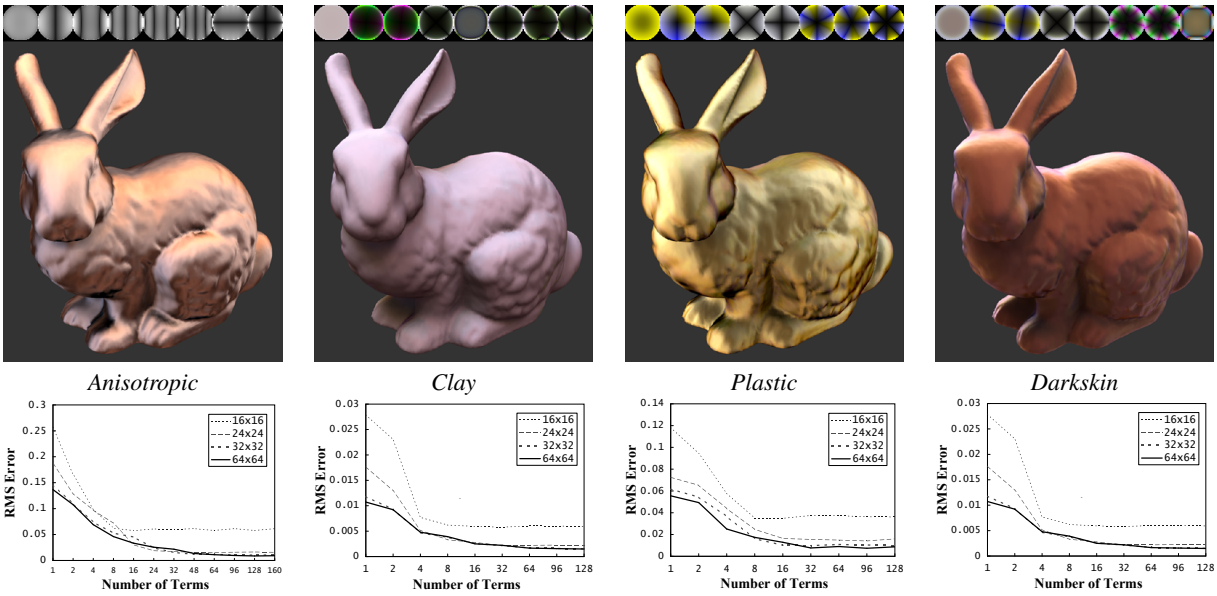


Figure 4: *BRDF decomposition results for four chosen BRDFs: the Ashikhmin-Shirley anisotropic BRDF, a light brown plastic BRDF, a Lafortune clay BRDF, and a Lafortune dark skin BRDF. The top images show the first 8 approximation terms, along with a rendering the the Bunny model using only the first 4 terms. Negative colors in the texture maps are displayed in absolute values. Notice that since our BRDF matrix is symmetric, the view maps differ with light maps only by a sign when the corresponding eigenvalue is negative, hence is not shown here. The bottom graphs plot the cosine-weighted RMS error of our BRDF approximation with increasing number of terms, and also with varying texture map resolution from* 16×16 *to* 64×64.

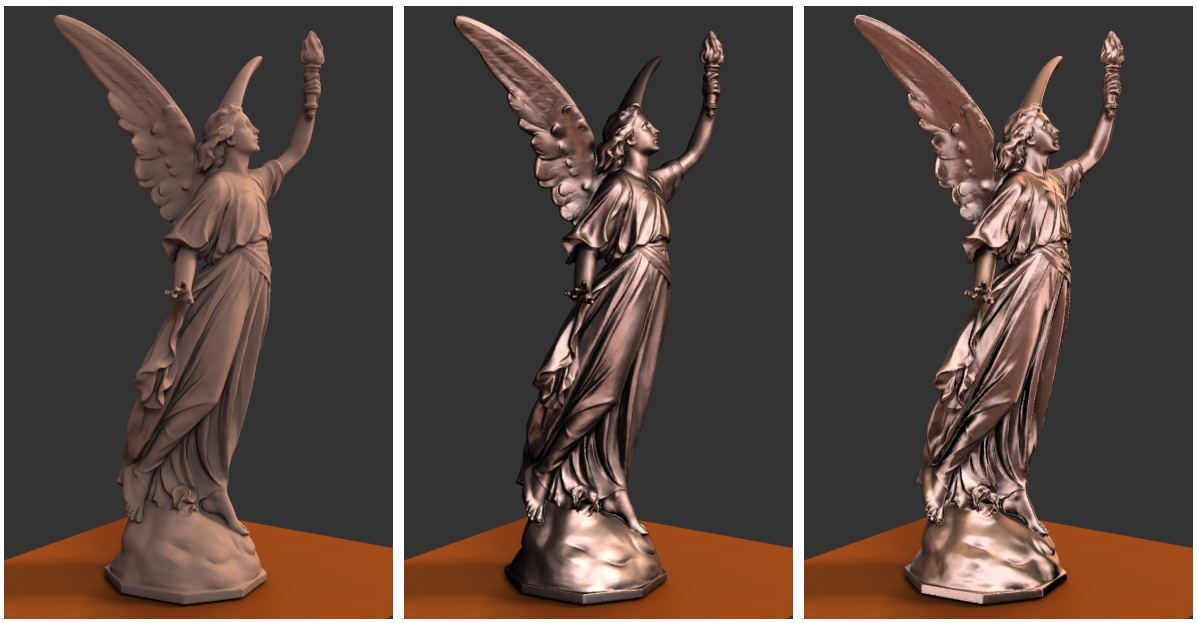


Figure 5: *From left to right, the Lucy model is rendered with a gray diffuse BRDF, a 1-term approximation (using polar coordinates parametrization in Eq 8) of the Ashikhmin-Shirley anisotropic BRDF and a 4-term approximation. Compare the middle image with the left image, we found a 1-term approximation is able to expose a reasonable amount of specular highlights.*

Figure 6: *Our technique is capable of capturing all-frequency illumination and shadows. In the left the Buddha model is illuminated by the* St. Peter's Basilica *environment map. Notice the clear definition of shadows on the ground. In the right the Lucy model is illuminated by a small area light, notice the contrast of Lucy's front face to its back face created by this high-frequency lighting.*

Figure 7: *The Lucy model rendered with a light brown plastic BRDF and illuminated by the* Eucalyptus Grove *environment map.*

efficient brdf representation. In *9th Eurographics Rendering Workshop*, pages 11–22, 1998. 3

[34] F. X. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg. A global illumination solution for general reflectance distributions. In *Proc. SIGGRAPH'91*, pages 187–196, 1991. 2

[35] P.-P. Sloan, J. Hall, J. Hart, and J. Snyder. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph.*, pages 382–391, 2003. 2, 7

[36] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proc. SIGGRAPH'02*, pages 527–536, 2002. 1, 2

[37] C. Soler and F. X. Sillion. Fast calculation of soft shadow textures using convolution. In *Proc. SIGGRAPH'98*, pages 321–332, 1998. 2

[38] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, 1997. 1

[39] S. H. Westin, J. R. Arvo, and K. E. Torrance. Predicting reflectance functions from complex surfaces. In *Proc. SIGGRAPH'92*, pages 255–264, 1992. 2

[40] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle. Surface light fields for 3d photography. In *Proc. SIGGRAPH'00*, pages 287–296, 2000. 1, 2

Appendix

The *power iteration* algorithm is commonly used to compute dominant matrix eigenvalues and eigenvectors. Since the eigenvectors of a real symmetric matrix $\mathbf{M}$ correspond to the column vectors of the left matrix $\mathbf{U}$ of $\mathbf{M}$'s SVD ($\mathbf{M} = \mathbf{USV}^T$), we can apply the power iteration algorithm on $\mathbf{M}$ to obtain its dominant singular values and vectors. In most cases it is fast and accurate enough for a low-order approximation of $\mathbf{M}$. The pseudocode is as follows:

- $\mathbf{M}$ is an $N \times N$ BRDF matrix
- K is the number of approximation terms
- $\mathbf{u}_k$ and $\mathbf{v}_k$ are the k-th view map and light map
- $\mathbf{M_R}$ is the residue BRDF matrix after each iteration

$\mathbf{M_R} = \mathbf{M}$
for $k = 1$ to K **do**
  $\mathbf{v}_k \leftarrow$ random $N \times 1$ non-zero vector
  repeat
    $\mathbf{v}_k \leftarrow \mathbf{M_R}\,\mathbf{v}_k$
    $\mathbf{v}_k \leftarrow \mathbf{v}_k / \| \mathbf{v}_k \|$ (normalization)
  until $\lambda_\mathbf{k} = \mathbf{v}_k^T\,\mathbf{M_R}\,\mathbf{v}_k / \| \mathbf{v}_k \|^2$ converges
  $\mathbf{v}_k \leftarrow \sqrt{|\lambda_\mathbf{k}|}\,\mathbf{v}_k$
  if $(\lambda_\mathbf{k} > 0)$ $\mathbf{u}_k \leftarrow \mathbf{v}_k$
  else $\mathbf{u}_k \leftarrow -\mathbf{v}_k$
  $\mathbf{M_R} = \mathbf{M_R} - \mathbf{u}_k\,\mathbf{v}_k^T$
end