To appear in SIGGRAPH 2003 Symposium on Interactive 3D Graphics

Matrix Radiance Transfer

Jaakko Lehtinen*
Remedy Entertainment, Ltd. and
Helsinki University of Technology,
Helsinki, Finland

Jan Kautz†
Max-Planck-Institut für Informatik,
Saarbrücken, Germany

Abstract

Precomputed Radiance Transfer allows interactive rendering of objects illuminated by low-frequency environment maps, including self-shadowing and interreflections. The expensive integration of incident lighting is partially precomputed and stored as matrices.

Incorporating anisotropic, glossy BRDFs into precomputed radiance transfer has been previously shown to be possible, but none of the previous methods offer real-time performance. We propose a new method, *matrix radiance transfer*, which significantly speeds up exit radiance computation and allows anisotropic BRDFs. We generalize the previous radiance transfer methods to work with a matrix representation of the BRDF and optimize exit radiance computation by expressing the exit radiance in a new, directionally locally supported basis set instead of the spherical harmonics. To determine exit radiance, our method performs four dot products per vertex in contrast to previous methods, where a full matrix-vector multiply is required. Image quality can be controlled by adapting the number of basis functions. We compress our radiance transfer matrices through principal component analysis (PCA). We show that it is possible to render directly from the PCA representation, which also enables the user to trade interactively between quality and speed.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation—Bitmap and frame buffer operations; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, Shading, Shadowing and Texture

Keywords: Shading, Reflectance & Shading Models, Spherical Harmonics, Orthogonal Projection.

1 Introduction

Lighting from area sources, soft shadows and interreflections are important effects in realistic image synthesis. Unfortunately, most methods for integrating over area light sources are too expensive for interactive rendering.

Recently, precomputed radiance transfer has been introduced as a means to shade objects with distant, low-frequency illumination, including self-shadowing and interreflections [Sloan et al. 2002]. This method is fast enough to achieve interactive and in certain cases even real-time rates. It only supports Phong-like [Phong 1975] glossy BRDFs, but can be extended to work with arbitrary spatially varying reflectance models [Kautz et al. 2002]. Both these techniques represent the precomputed transfer as well as the BRDF in spherical harmonics (SH). The method by Sloan *et al.* [2002] represents the BRDF as a 2D filter kernel, whereas Kautz *et al.* [2002] model the BRDF by view-dependent SH coefficients.

We take another approach and achieve higher performance with less memory consumption. Our contributions are:

Efficient evaluation of exit radiance. We express exit radiance at the vertices in a directionally compactly supported basis instead of the spherical harmonics. Our method does not restrict the choice of the basis functions, and particularly does not require the basis to be orthonormal; we use a collection of piecewise bilinear functions defined on the hemisphere above each vertex. In order to determine exit radiance into the viewing direction, only four of these basis functions' coefficients need to be determined; thus only four dot products are computed for each vertex. This is in contrast with the previous methods, where a full matrix multiply is always needed. The basis change decouples the number of terms used in the precomputed transfer simulation and the BRDF representation from the runtime workload. We also analyze the error introduced by the change of basis; this information can be used for guiding decisions on the number of basis functions to be used.

Matrix BRDF representation. To allow general, anisotropic BRDFs, we use the matrix representation of Westin *et al.* [1992]. With BRDFs represented as matrices, we express the whole chain from incident lighting (in SH) to exit radiance (in the directional basis) by a matrix operating on the incident lighting's SH coefficients.

Compression. We reduce memory consumption to a practical level by applying principal component analysis (PCA) to the radiance transfer matrices. We achieve a 1:25 compression ratio with visual results very close to uncompressed results; greater compression ratios of above 1:100 produce pleasing results for non-self-transferred models. We show that it is possible to render directly from the PCA representation. Runtime adjustment of the number of principal components used provides fine-grained quality/speed tradeoff.

2 Related Work

This work allows rendering of glossy reflections from rigid objects illuminated by environment maps, incorporating effects such as self-shadowing and interreflections. Related work can be found in three areas: environment mapping, precomputed transfer, and spherical harmonics for shading. We briefly summarize previous work in these areas.

The *environment map* technique for rendering mirror-like reflections on curved objects was first introduced by Blinn and Newell [1976]. Greene [1986a; 1986b] observed that a pre-convolved environment map could be used for simulating diffuse and glossy reflections. Instead of storing the incident radiance, Greene simply stored exit radiance.

Since then, several approaches have been proposed for simulating glossy reflections based on pre-filtered environment maps. These algorithms either assume a simple, fixed BRDF

*jaakko@tml.hut.fi
†kautz@mpi-sb.mpg.de

model [Greene 1986a; Greene 1986b; Heidrich and Seidel 1999; Kautz et al. 2000; McAllister et al. 2002; Ramamoorthi and Hanrahan 2001b] or generalize to isotropic BRDFs [Cabral et al. 1999; Kautz and McCool 2000; Latta and Kolb 2002; Ramamoorthi and Hanrahan 2002]. Some of these techniques [Kautz et al. 2000; Ramamoorthi and Hanrahan 2001b; Ramamoorthi and Hanrahan 2002] can handle dynamic illumination, but none of them can handle spatially varying BRDFs, self-shadowing and interreflections.

Precomputed transfer for micro-geometry has been proposed in various ways. Heidrich *et al.* [2000] generalized horizon mapping [Max 1988] to diffuse and glossy interreflections, though changes due to dynamic lighting were not quite real-time. Polynomial texture maps [Malzbender et al. 2001] allow real-time but view-independent interreflection effects as well as shadowing. A similar approach using a steerable basis for directional lighting was used by Ashikhmin and Shirley [2002]. These methods precompute a simple representation of transfer, but are only valid for directional light sources, thus requiring multiple integration in order to simulate area light sources. The work by Sloan *et al.* [2002] and by Kautz *et al.* [2002] will be reviewed in more detail in the next section.

Spherical harmonics (SH) [Edmonds 1960] have often been utilized in computer graphics, as they have properties similar to the Fourier basis, but over the unit sphere, and are thus well suited for representing bandlimited spherical functions. Cabral *et al.* [1987] utilized spherical harmonics to derive isotropic BRDFs from heightfields and made the observation that their use reduces the lighting integral to a dot product. Kautz *et al.* [2002] used this insight to extend precomputed radiance transfer to arbitrary BRDFs. Westin *et al.* [1992] used spherical harmonics for off-line BRDF inference from geometric models. In their method both view and light dependence of the BRDF are represented as a large matrix using the spherical harmonics basis. In our work we make use of this matrix representation. The previously mentioned environment map techniques by Ramamoorthi and Hanrahan [2001b; 2002] are also based on spherical harmonics.

3 Precomputed Radiance Transfer

Here we review the original techniques of Sloan *et al.* [2002] and Kautz *et al.* [2002], since our new method is based on their work. Both techniques store a matrix representation of self-transfer directly over the object's vertices.

Sloan *et al.* assume that the object is illuminated by distant low-frequency illumination $L_{in}(\hat{l})$, represented for example by an environment map. To compute exit radiance from a point p on an object's surface, the integral

$$\begin{aligned} L_{out,p}(\hat{v}) &= \int_\Omega L_{in}(\hat{l})\, V_p(\hat{l})\, f_r(\hat{v},\hat{l})\, \max(0,\hat{n}\cdot\hat{l})\, d\hat{l} = \\ &= \int_\Omega L_{in}(\hat{l})\, V_p(\hat{l})\, f_r^*(\hat{v},\hat{l})\, d\hat{l} \end{aligned} \quad (1)$$

needs to be evaluated at each p, where $\hat{n}$ is the surface normal at p, $L_{in}(\hat{l})$ is the incident radiance, $V_p(\hat{l})$ is the visibility function — zero for directions along which the environment cannot be seen due to self-shadowing and one if the environment can be seen — and $f_r^*(\hat{v},\hat{l})$ is the reflectance model including the cosine term. The result $L_{out,p}$ is the radiance leaving the point p to the direction $\hat{v}$, properly attenuated by self-shadowing.

3.1 Transferred Radiance

In the general case the integral in Equation (1) is expensive to compute, since the visibility function $V_p(\hat{l})$ changes at every point on the object. Fortunately, under the assumption that the object is rigid, this function remains constant for each point, and thus has to be computed only once. Furthermore, if the incident lighting is represented as a coefficient vector in the spherical harmonics basis,

Variable	Meaning
$\hat{v}$, $\hat{v}_\omega$	viewing direction (global/local)
$\hat{r}_{\hat{v}}$, $\hat{r}_{\hat{v}_\omega}$	reflected viewing direction (global/local)
$\hat{l}$, $\hat{l}_\omega$	light direction (global/local)
L_{in} or L	incident radiance
L_{out} or L_o	exit radiance
$V_p(\hat{l})$	visibility function
$L_p^*(\hat{l})$	transferred radiance, in global coordinates
$L_p'(\hat{l})$	transferred radiance, in local coordinates
$f_r(\hat{l},\hat{v})$	BRDF
$f_r^*(\hat{l},\hat{v})$	BRDF product function, $f_r(\hat{l},\hat{v})\max(0,\hat{n}\cdot\hat{l})$
$f_r^*(\hat{l}_\omega,\hat{v}_\omega)$	BRDF product function, in local coordinates
M^2	number of new basis functions
m	order of exit radiance SH expansion
n	order of incident lighting SH expansion
$\mathcal{B}$	BRDF matrix, $m^2 \times n^2$
$\mathcal{C}$	change of basis matrix, $M^2 \times m^2$
$\mathcal{G}$	Gram matrix, $M^2 \times M^2$
$\mathcal{T}_p$	transfer matrix for vertex p, $n^2 \times n^2$
$\mathcal{R}_p$	SH rotation matrix (global to local coordinates)
Y_i	spherical harmonic functions
g_j	basis functions with small directional supports

Figure 1: *List of used variables/terms.*

the *transferred radiance* $L_p^*(\hat{l}) = L_{in}(\hat{l})V_p(\hat{l})$, or more precisely its coefficient vector in the SH basis, can be computed with a matrix-vector multiplication from the SH projection coefficients $\vec{L}$ of the incident lighting. This *transfer matrix* $\mathcal{T}_p$, which varies with the surface location p, can also be extended to include interreflection effects in addition to self-shadowing when determining transferred radiance [Sloan et al. 2002]. The transferred radiance is often required to be represented in the local tangent space of a vertex; this can be achieved by multiplication with a high-dimensional rotation matrix $\mathcal{R}_p$. We denote the transferred radiance SH vector in local coordinates by $\vec{L}_p'$.

3.2 Exit Radiance

Three methods have been proposed for the remaining task of computation of exit radiance by integrating the transferred radiance against the BRDF. These methods make different assumptions about the BRDF.

Diffuse BRDF. If the BRDF is ideally diffuse, a single dot product is required per vertex to evaluate the exit radiance [Sloan et al. 2002].

Phong-like BRDF: Integration via Convolution. If the BRDF is symmetric with respect to the local reflected view direction $\hat{r}_{\hat{v}}$, which is the case for e.g. the Phong [1975] model, the computation of the integral simplifies to a spherical convolution by a BRDF kernel, followed by evaluation of an SH expansion in the reflected viewing direction $\hat{r}_{\hat{v}_\omega}$ [Sloan et al. 2002]. Because the BRDF convolution kernel is also represented in spherical harmonics, the convolution operation becomes a scaled component-wise multiplication between the transferred radiance SH coefficients $\vec{L}_p'$ and the filter kernel's coefficients [Ramamoorthi and Hanrahan 2001a]. The operation results in an SH coefficient vector for the exit radiance. To obtain actual exit radiance, the SH expansion must be evaluated at the local reflection direction $\hat{r}_{\hat{v}_\omega}$.

General BRDF: Integration via Projection. The method of Kautz *et al.* [2002] allows arbitrary, anisotropic BRDFs. They reparametrize f_r^* by the local viewing direction $\hat{v}_\omega$ to get a spherical function $f_v(\hat{l}_\omega)$ for each $\hat{v}_\omega$. The functions $f_v(\hat{l}_\omega)$ are projected into the SH basis, yielding view-dependent SH coefficients $f_i(\hat{v}_\omega)$. The coefficients are stored in a parabolic texture map indexed by the viewing direction. For each vertex, exit radiance is computed by performing a dot product of the view-dependent BRDF coeffi-

cient $f_i(\hat{v}_\omega)$ with the incident illumination coefficients. The BRDF f^* is represented in the local surface frame, which varies over the object while the incident lighting uses a global coordinate system. The two coordinate systems must be aligned at every vertex by rotating the incident lighting into the local coordinate frame with a high-dimensional rotation matrix.

3.3 Summary

In both of the algorithms for glossy reflection discussed in the previous section, the following steps need to be taken. First the per-vertex transfer matrices are precomputed off-line. At run-time the incident lighting is projected into the SH basis. After projection, the incident lighting is multiplied with the transfer matrix at each vertex to get transferred radiance. The method of Sloan *et al.* computes a pair-wise multiplication of transferred radiance with the BRDF kernel, followed by an SH evaluation in the reflected viewing direction. The method of Kautz *et al.* first rotates the transferred radiance into the local tangent frame of each vertex and then computes a dot product with the view-dependent BRDF coefficients, directly resulting in exit radiance.

Rendering can be sped up by fixing the lighting, because then transferred radiance can be precomputed (and also prerotated), eliminating the matrix-vector multiplication at run-time. If the view is fixed, a similar speed up can be achieved [Kautz et al. 2002].

Our goal is to speed up the computation for the general case, where both the lighting and the view can change simultaneously.

4 Matrix Radiance Transfer

In this section we explain how the computation of exit radiance is made more efficient by expressing exit radiance in a directionally locally supported basis. We also present a new way of integrating the transferred radiance against the BRDF product function; to this end we utilize a matrix representation for the BRDF. Using these results, we show how precomputed transfer with arbitrary, spatially varying BRDFs and efficient exit radiance representation can be cast into a general matrix expression.

4.1 Transfer

In our framework, the incident lighting is projected into the SH basis, forming a coefficient vector $\vec{L}$. At each vertex, this coefficient vector is transformed into transferred radiance by multiplication with the transfer matrix $\mathcal{T}_p$, which may taken to be the identity transformation if self-shadowing and interreflections are not required.

4.2 BRDF Matrix

We use the method of Westin *et al.* [1992] for representing BRDFs as matrices — we double-project the 4D BRDF product function $f_r^*(\hat{l}_\omega, \hat{v}_\omega)$ into the SH basis. This is achieved by first projecting the view-dependence of the BRDF, followed by the light-dependence. This results in a matrix representation for the general, anisotropic BRDF. We briefly reiterate the development here.

For each incident direction $\hat{l}_\omega$, the BRDF product function is a function over the unit hemisphere. If this function is projected into the SH basis $Y_i(\hat{v}_\omega)$ utilizing a suitable extension [Westin et al. 1992] to the whole unit sphere, we get

$$f_r^*(\hat{l}_\omega, \hat{v}_\omega) \approx \sum_{i=1}^{m^2} b_i(\hat{l}_\omega) Y_i(\hat{v}_\omega), \tag{2}$$

with the coefficient vectors b_i depending on the incident direction. Since the projection coefficients $b_i(\hat{l}_\omega)$ are functions over the hemisphere, we project the coefficients themselves into SH:

$$b_i(\hat{l}_\omega) \approx \sum_{j=1}^{n^2} b_{ij} Y_j(\hat{l}_\omega). \tag{3}$$

Hence, the final doubly-projected form for the BRDF product function is

$$f_r^*(\hat{l}_\omega, \hat{v}_\omega) \approx \sum_{i=1}^{m^2} \left(\sum_{j=1}^{n^2} b_{ij} Y_j(\hat{l}_\omega) \right) Y_i(\hat{v}_\omega). \tag{4}$$

We denote the matrix with elements b_{ij} by $\mathcal{B}$.

Since the transferred, rotated radiance $\vec{L'}_p$ is also represented by SH coefficients, we can substitute its SH expansion and the BRDF expansion from Equation (4) into the lighting Equation (1):

$$L_{out,p}(\hat{v}_\omega) = \int_\Omega \left[\sum_{i=1}^{n^2} L'_{p,i} Y_i(\hat{l}_\omega) \right] \left[\sum_{j=1}^{m^2} \left(\sum_{k=1}^{n^2} b_{p,jk} Y_k(\hat{l}_\omega) \right) Y_j(\hat{v}_\omega) \right] d\hat{l}_\omega,$$

from where by reordering the summations and integration and using the orthonormality of the SH basis we get the vector of coefficients of full spherical outgoing radiance, expressed in m^2 SH coefficients:

$$\begin{aligned} \vec{L}_{out,p} &= \mathcal{B}_p \cdot \vec{L}'_p \\ &= \mathcal{B}_p \mathcal{R}_p \mathcal{T}_p \cdot \vec{L}. \end{aligned} \tag{5}$$

This means that for each vertex, the incident lighting $\vec{L}$ is first transformed by the transfer matrix, resulting in transferred radiance, which is then rotated into the local tangent space of vertex p. The local transferred radiance is then multiplied with the location-dependent BRDF matrix $\mathcal{B}_p$, resulting in a SH coefficient vector representing the exit radiance function $L_{out,p}(\hat{v}_\omega)$ at the vertex p. Finally, this SH expansion needs to be evaluated for the actual viewing direction $\hat{v}_\omega$ to get a final exit radiance value:

$$L_{out,p}(\hat{v}_\omega) = \sum_{i=1}^{m^2} L_{out,p,i} Y_i(\hat{v}_\omega). \tag{6}$$

4.3 Change of Basis

To compute exit radiance using Equation (6), a matrix-vector multiplication is necessary regardless of the viewing direction $\hat{v}_\omega$; all the SH coefficients $\vec{L}_{out,p}$ are required, since the SH basis functions have global support over the sphere. On the other hand, if exit radiance would be expressed in a basis set with local supports, one would only need to evaluate the basis functions "close" to the viewing direction. Motivated by this observation, we project the exit radiance into a new basis, the span of a set of functions locally supported on the hemisphere above each vertex.

The exit radiance function at point p, given by Equation (6), represented in the SH basis by coefficients $\vec{L}_{out,p}$, can be transformed into an expansion in another basis function set $g_j(\hat{v}_\omega)$, with $j = 1, \ldots, M^2$, with coefficients $\vec{L}^g_{out,p}$. We choose to perform this transformation by orthogonal projection [Kreyszig 1989], which is a linear transformation, and can thus be represented by a matrix operating on the coefficients $\vec{L}_{out,p}$. In addition, orthogonal projection has the property of minimizing the transformation error in the L_2 norm. The orthogonal projection matrix has the form $\mathcal{G}^{-1}\mathcal{C}$, where the elements of the $M^2 \times m^2$ change of basis matrix $\mathcal{C}$ and those of the symmetric $M^2 \times M^2$ matrix $\mathcal{G}$ are defined as

$$\mathcal{C}_{ji} = \int_\Omega Y_i(\hat{\omega}) g_j(\hat{\omega}) d\hat{\omega}, \quad \mathcal{G}_{ji} = \int_\Omega g_i(\hat{\omega}) g_j(\hat{\omega}) d\hat{\omega}. \tag{7}$$

Now the coefficients of the exit radiance function expressed in the new basis are given by

$$\vec{L}^g_{out,p} = \mathcal{G}^{-1} \mathcal{C} \vec{L}_{out,p}. \tag{8}$$

Please refer to Appendix A for a full derivation of the basis change matrices.

In our implementation we use a set of $M \times M$ piecewise bilinear functions ("tent" functions centered at grid points) defined over the unit square as the new basis functions g_j. The basis functions are mapped from the unit square onto the unit hemisphere by an area-preserving bijection [Shirley and Chiu 1997, p.6]. Due to the properties of the piecewise bilinear functions, only four basis functions have a nonzero value for any given direction on the unit hemisphere. The bilinear functions also have simple expressions.

4.4 Full Formulation

Combining the results from the previous sections we get the following equation for exit radiance at a vertex p on the object's surface:

$$\begin{aligned} L^g_{out,p}(\hat{v}_\omega) &= \sum_{j=1}^{M^2} g_j(\hat{v}_\omega) \left(\mathcal{G}^{-1}\, \mathcal{C}\, \mathcal{B}_p\, \mathcal{R}_p\, \mathcal{T}_p \cdot \vec{L} \right)_j \\ &= \sum_{j=1}^{M^2} g_j(\hat{v}_\omega) \left(\mathcal{A}_p \cdot \vec{L} \right)_j . \end{aligned} \tag{9}$$

This equation shows that we first have to compute a linear transformation on the SH coefficient vector $\vec{L}$ of the incident lighting. The result of this linear transformation is the spherical function representing outgoing radiance, expressed as a coefficient vector in the new basis spanned by the functions g_j. This spherical function is then evaluated by multiplying the transformed coefficient vector with the values of the basis functions g_i evaluated at direction $\hat{v}_\omega$ and summing the results.

Since we have chosen a set of locally supported basis functions, it is not necessary to compute the full coefficient vector $\vec{L}^g_{out,p}$, but only the coefficients $L^g_{out,p,k}$, for which the basis functions $g_k(\hat{v}_\omega)$ are nonzero at $\hat{v}_\omega$. The piecewise bilinear basis functions have the property that only four $g_k(\hat{v}_\omega)$ are non-zero for any given $\hat{v}_\omega$; this means that only four coefficients $L^g_{out,p,k}$ need to be computed, and thus the number of multiplication operations per vertex only depends on n^2, the number of coefficients used for representing incident lighting; our representation decouples the order of the BRDF representation (m) and the total number of directional basis functions (M^2) from the computational complexity of exit radiance determination.

Changing the spherical harmonics basis at the end instead of representing incident lighting and performing all the computations in the new basis has several advantages. Firstly, projection of the incident lighting into SH is fast, since the basis functions are orthonormal. Furthermore, the alignment of the incident lighting's coordinate system with the vertices' local coordinate systems can be done exactly and without aliasing.

5 Analysis of the Basis Change

To analyze the error introduced by the basis change, we examine the average PSNR of exit radiance expansions projected into the $M \times M$ piecewise bilinear basis compared to the corresponding original m-th order SH expansions with m^2 coefficients. A quadratic expression of the form

$$\|L_o - L^g_o\|_2^2 = \vec{L}_o^T\, \mathcal{E}\, \vec{L}_o, \tag{10}$$

where $\vec{L}_o$ is the SH coefficient vector for L_o and $\mathcal{E}$ is an $m^2 \times m^2$ matrix, can be derived by expanding out the squared norm as $\int_\Omega (L_o - L^g_o)^2 d\omega = \langle L_o - L^g_o, L_o - L^g_o \rangle$. The derivation is straightforward and we omit the details. Equation (10) allows to directly compute the squared norm using only $\vec{L}_o$ and $\mathcal{E}$, instead of computing it using hemispherical sampling. Average PSNR can then be measured as

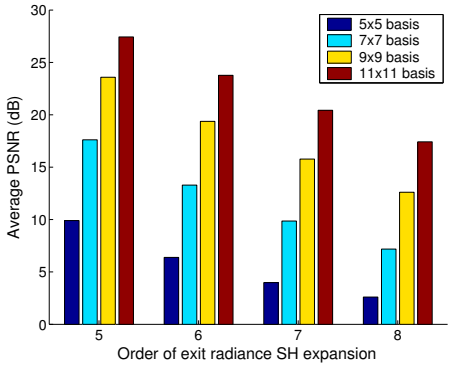


Figure 2: ***Error Analysis***. *Order of exit radiance SH epansion m vs. average PSNR of bilinear projections of random SH expansions.*

$$PSNR_{avg} = -10 \log_{10} \frac{1}{E\{\|L_o - L^g_o\|_2^2\}}, \tag{11}$$

with the expected value taken from uniformly distributed random norm-1 coefficient vectors $\vec{L}_o$. Figure 2 shows average PSNRs measured from populations of 1000 coefficient vectors for different values of m and M. Poor PSNRs resulting from use of too few basis functions do not directly translate to unusable image quality but rather to lack of detail, as the error in the projection is coherent; see Figure 4 for an example.

6 Matrix Compression

For each vertex p on the object, $\mathcal{A}_p$ is an $M^2 \times n^2$ matrix, where n is the SH order of the incident lighting and M^2 is the number of exit radiance basis functions; $n = 5$ and $M = 11$ are representative values. For an object with 50 000 vertices, the total amount of data is approximately 600 MB.

We use principal component analysis (PCA) [Gonzales and Woods 1993] to compress the data. We apply PCA directly to the matrices $\mathcal{A}_p$.[1] This leads to the following representation:

$$\mathcal{A}_p \approx \sum_{k=1}^{K} w_{p,k}\, \mathcal{A}^{PCA}_k, \tag{12}$$

where every matrix $\mathcal{A}_p$ is now represented as a weighted sum of basis matrices $\mathcal{A}^{PCA}_k$ with varying weights $w_{p,k}$ for each vertex. K, an integer between 1 and $M^2 n^2$, is the number of principal components.

It is possible to render directly from the PCA-compressed representation; to determine the exit radiance for each vertex, we need to evaluate

$$L^g_{out,p}(\hat{v}_\omega) \approx \sum_{j=s,t,r,o} g_j(\hat{v}_\omega) \left(\sum_{k=1}^{K} w_{p,k}\, \mathcal{A}^{PCA}_k \cdot \vec{L} \right)_j , \tag{13}$$

where s,t,r and o are the indices of the basis functions that are nonzero for $\hat{v}_\omega$. With $\vec{L}^{PCA}_k := \mathcal{A}^{PCA}_k \cdot \vec{L}$, one can see that the $\vec{L}^{PCA}_k$ can be computed once and reused for all vertices. Furthermore, since only four components of the $\vec{L}^{PCA}_k$ are needed for each vertex, the exit radiance expansion coefficients can be found by $4K$ multiplications per color channel per vertex.

[1] We write the matrices $\mathcal{A}_p$ as column vectors of length $M^2 n^2$ by stacking their columns and concatenate the vectors horizontally to get a matrix of size $M^2 n^2 \times V$, where V is the vertex count. After applying PCA to this matrix, the principal components are reorganized back into matrices $\mathcal{A}^{PCA}_k$.

Quality of the compression can be controlled by adjusting K. Figure 3 shows average compression errors for different K, measured as the average PSNR of exit radiance expansions from compressed vs. uncompressed matrices. The computation of PSNRs is done similarly to the analysis of the basis change in section 5. An Ashikhmin-Shirley BRDF was used, with $N_{u,v} = 40$ for the specular and $N_{u,v} = 5$ for the diffuse case. The graph shows that self-transfer effects and glossiness of the BRDF decrease the quality achieved by PCA compression.

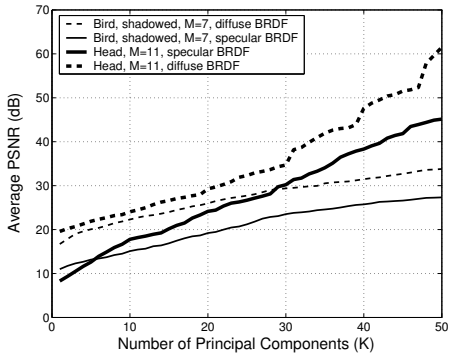


Figure 3: ***PCA Quality***. *With each matrix $\mathcal{A}_p$ approximated by Equation (12) as $\mathcal{A}_p^{PCA}$, the error is measured as the average PSNR of the exit radiance expansions produced by compressed matrices with different number of principal components compared to uncompressed matrices. Here $m = 5$.*

7 Results

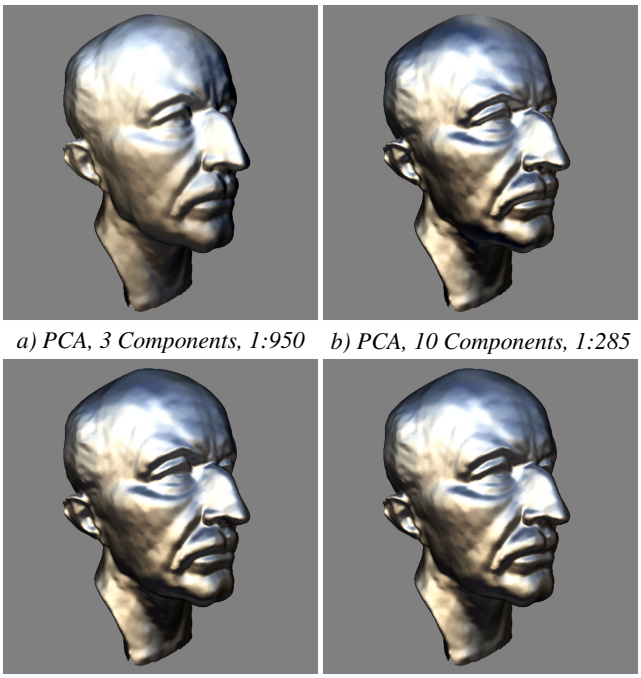
Figure 4 shows an object with the Ashikhmin-Shirley BRDF [Ashikhmin and Shirley 2000] rendered using different methods. The BRDF matrix representation makes points near object silhouettes slightly darker compared to the method of Kautz *et al.* [2002]; this can be alleviated using a larger m in the BRDF approximation at the cost of having to also increase M. We use $m = 8$ for all images in this paper, with empirically-determined windowing [Westin et al. 1992] in the higher-order SH bands (> 5) to avoid ringing. The change of basis produces visually pleasing results for $M >= 9$, with differences to the direct evaluation of Equation (6) only noticeable by close examination. Less basis functions may be used if m is smaller.

Table 1 summarizes timings for our method. The figures include only the calculation of lighting for the vertices, i.e. actual picture generation is excluded. All methods are implemented in software to allow fair comparison of the algorithms. 25 SH coefficients were used for incident lighting in all cases. As expected, the change of basis is faster than the previous method by a factor of 3 to 6, although the speedup achieved without PCA compression is not quite as big as looking at operations counts per vertex alone would suggest. We explain this with cache misses, as a lot of memory is accessed each frame. On the other hand, the rates achieved by rendering directly from the PCA representation are much closer to theoretical estimates. Also, the speed of the PCA renderer depends on K, which can be changed at run-time. With a lower K we achieve render rates significantly higher than the other methods.

Figure 5 shows renderings with and without compression for different values for K using 11×11 directional basis functions. As expected, using more principal components achieves better quality. With $K > 25$, visual difference in non-self-transferred models is in practice noticeable only near singularities of the tangent field, where the possible anisotropy of the BRDF is most evident. Models with self-transfer require more principal components to converge visually; we have found $K = 50$ produces good results at faster rates than when not using PCA. Using less components loses accuracy gracefully.

model	vertices	a) fps	b) fps	c) fps	d) fps	e) fps
bird	50k	1.6	9.3	8.7	28.9	17.1
head	50k	2.9	9.7	8.7	26.6	17.6

Table 1: ***Timings***. *All timings were measured from a software implementation running on an Intel P4 2.26 GHz with 1 GB of DDR266 main memory. The bird model was rendered with self-shadowing and the head model without. The timings are for lighting only. a) Kautz* et al.*, b) Basis change 7x7 without PCA, c) Basis change 11x11 without PCA, d) Basis change 11x11, $K = 10$ (1:285 compression), e) Basis change 11x11, $K = 25$ (1:114 compression).*



a) PCA, 3 Components, 1:950 *b) PCA, 10 Components, 1:285*

c) PCA, 20 Components, 1:142 *d) Uncompressed*

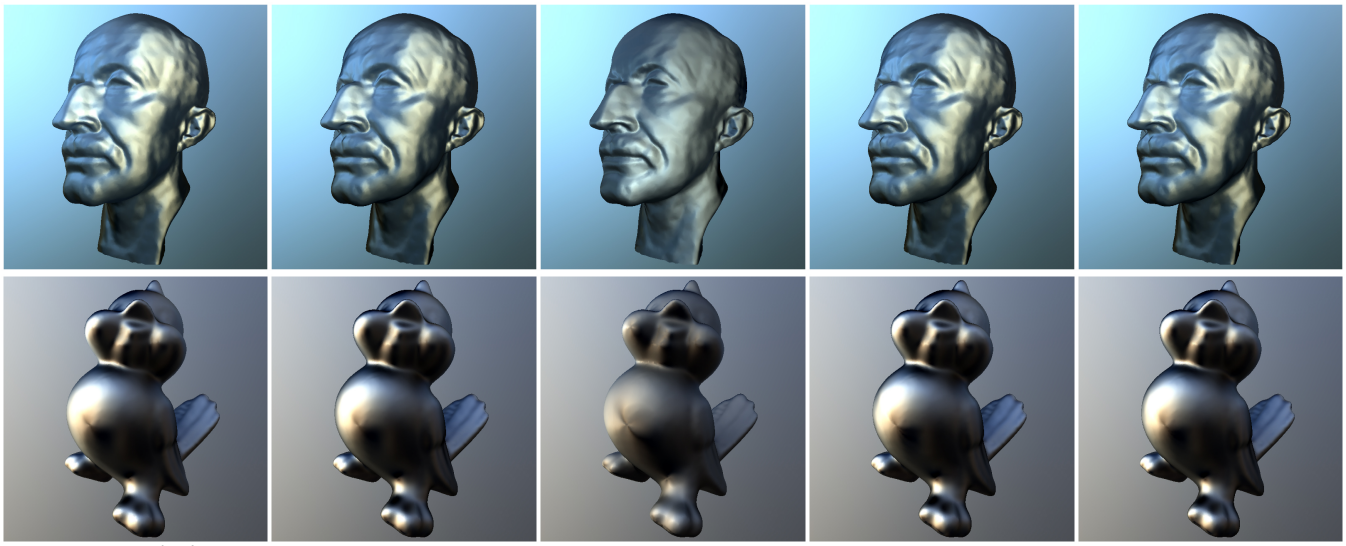
Figure 5: ***PCA Comparison***. *Effect of PCA compression on a change into an 11×11 basis with $m = 8$. The reconstructed results and compression ratios are shown for different values of K. The model is rendered without self-transfer.*

8 Conclusions and Future Work

We have presented a method that allows rendering of objects illuminated by distant, low-frequency lighting with self-shadowing and interreflection effects. Our method fully supports arbitrary viewpoints and time-varying lighting and is faster than previous techniques [Kautz et al. 2002; Sloan et al. 2002].

The speedup is achieved by expressing exit radiance from the vertices of the object in a new, directionally locally supported function basis. This reduces the large per-vertex matrix-vector multiplication required by previous methods to four dot products and removes the need for per-vertex SH function evaluations. We also showed how a BRDF matrix representation [Westin et al. 1992] — which allows anisotropic BRDFs — can be included into the precomputed radiance transfer framework. This enables us to express the whole transformation from SH-projected incident irradiance to exit radiance by a single matrix per vertex. Our method also has the advantage that we can use a different BRDF at each point on the object with no additional memory cost. Admittedly, our BRDF matrices need to be quite large in order to represent high-frequency BRDFs.

We compress the resulting large matrix data set by principal component analysis (PCA). The compression reduces memory consumption to a practical level and can be used to trade speed vs. quality at runtime.



a) Previous method [Kautz et al. 2002] *b) Equation (6)* *c) Basis Change* 3×3 *d) Basis Change* 9×9 *e) Basis Change* 11×11

Figure 4: ***A Comparison of Methods***. *We compare the previous method and our new method with and without change of basis. The head model is rendered without selfshadowing, using an isotropic Ashikhmin-Shirley BRDF ($N_{u,v} = 40$). The bird model is rendered with self-shadowing and an anisotropic Ashikhmin-Shirley BRDF ($N_u = 40, N_v = 10$). 3×3 basis change has been included as an example of using too few basis functions. The images in column b) are rendered by directly evaluating Equation (6).*

In the future we would like to implement our new method on graphics hardware. Furthermore, we would like to try other compression schemes, which hopefully increase run-time performance even more and perform better with complex self-transfer effects. Deformable models cannot be handled by our method; we would like to extend our method to support them as well.

9 Acknowledgements

We thank Jussi Räsänen for the code platform on which our implementation is based, Paul Debevec (www.debevec.org) for the lighting environments [Debevec 1998], and all friends, colleagues and the anonymous reviewers for helpful comments and criticism.

References

ASHIKHMIN, M., AND SHIRLEY, P. 2000. An Anisotropic Phong BRDF Model. *Journal of Graphics Tools 2*, 5, 25–32.

ASHIKHMIN, M., AND SHIRLEY, P. 2002. Steerable Illumination Textures. *ACM Transactions on Graphics 21*, 1, 1–19.

BLINN, J., AND NEWELL, M. 1976. Texture and Reflection in Computer Generated Images. *Communications of the ACM 19*, 542–546.

CABRAL, B., MAX, N., AND SPRINGMEYER, R. 1987. Bidirectional Reflection Functions From Surface Bump Maps. In *Proceedings of ACM SIGGRAPH 87*, 273–281.

CABRAL, B., OLANO, M., AND NEMEC, P. 1999. Reflection Space Image Based Rendering. In *Proceedings of ACM SIGGRAPH 99*, 165–170.

DEBEVEC, P. 1998. Rendering Synthetic Objects Into Real Scenes: Bridging Traditional and Image-Based Graphics With Global Illumination and High Dynamic Range Photography. In *Proceedings of ACM SIGGRAPH 98*, 189–198.

EDMONDS, A. 1960. *Angular Momentum in Quantum Mechanics*. Princeton University, Princeton, NJ.

GONZALES, R. C., AND WOODS, R. E. 1993. *Digital Image Processing*. Addison-Wesley.

GREENE, N. 1986. Applications of World Projections. In *Proceedings Graphics Interface*, 108–114.

GREENE, N. 1986. Environment Mapping and Other Applications of World Projections. *IEEE Computer Graphics & Applications 6*, 11, 21–29.

HEIDRICH, W., AND SEIDEL, H. 1999. Realistic, Hardware-accelerated Shading and Lighting. In *Proceedings of ACM SIGGRAPH 99*, 171–178.

HEIDRICH, W., DAUBERT, K., KAUTZ, J., AND SEIDEL, H.-P. 2000. Illuminating Micro Geometry Based on Precomputed Visibility. In *Proceedings of ACM SIGGRAPH 2000*, 455–464.

KAUTZ, J., AND MCCOOL, M. 2000. Approximation of Glossy Reflection with Prefiltered Environment Maps. In *Proceedings Graphics Interface*, 119–126.

KAUTZ, J., VÁZQUEZ, P.-P., HEIDRICH, W., AND SEIDEL, H.-P. 2000. A Unified Approach to Prefiltered Environment Maps. In *Eleventh Eurographics Workshop on Rendering*, 185–196.

KAUTZ, J., SLOAN, P.-P., AND SNYDER, J. 2002. Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics. In *13th Eurographics Workshop on Rendering*, 301–308.

KREYSZIG, E. 1989. *Introductory Functional Analysis with Applications*. Wiley, New York.

LATTA, L., AND KOLB, A. 2002. Homomorphic Factorization of BRDF-based Lighting Computation. *ACM Transactions on Graphics 21*, 3 (July), 509–516.

MALZBENDER, T., GELB, D., AND WOLTERS, H. 2001. Polynomial texture maps. In *Proceedings of ACM SIGGRAPH 2001*, 519–528.

MAX, N. 1988. Horizon Mapping: Shadows for Bump-Mapped Surfaces. *The Visual Computer 4*, 2 (July), 109–117.

MCALLISTER, D., LASTRA, A., AND HEIDRICH, W. 2002. Efficient Rendering of Spatial Bi-directional Reflectance Distribution Functions. In *Proceedings Graphics Hardware*, 79–88.

PHONG, B.-T. 1975. Illumination for Computer Generated Pictures. *Communications of the ACM 18*, 6 (June), 311–317.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. A Signal-Processing Framework for Inverse Rendering. In *Proceedings of ACM SIGGRAPH 2001*, 117–128.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of ACM SIGGRAPH 2001*, 497–500.

RAMAMOORTHI, R., AND HANRAHAN, P. 2002. Frequency Space Environment Map Rendering. *ACM Transactions on Graphics 21*, 3, 517–526.

SHIRLEY, P., AND CHIU, K. 1997. A low distortion map between disk and square. *Journal of Graphics Tools 2*, 3.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics 21*, 3, 527–536.

WESTIN, S., ARVO, J., AND TORRANCE, K. 1992. Predicting Reflectance Functions From Complex Surfaces. In *Proceedings of ACM SIGGRAPH 92*, 255–264.

Appendix A Orthogonal Projection

By definition [Kreyszig 1989], the orthogonal projection Pf of a spherical function $f(\omega)$ onto the span of M^2 linearly independent functions $g_i(\omega)$ is characterised by $\langle f - Pf, g_i \rangle = 0$ for all i; that is, the "approximation error" is required to be orthogonal to the target basis. The problem is to find the coefficients l_i^g for the representation $Pf = \sum_{i=1}^{M^2} l_i^g \, g_i(\omega)$. Expanding the orthogonality requirement for each $k = 1, \ldots, M^2$ we get

$$0 = \langle g_k, f - Pf \rangle = \left\langle g_k, f - \sum_{i=1}^{M^2} l_i^g \, g_i \right\rangle = \langle g_k, f \rangle - \sum_{i=1}^{M^2} l_i^g \, \langle g_k, g_i \rangle \,, \tag{14}$$

where we have used $\langle X, Y \rangle = \int_\Omega X(\omega) Y(\omega) d\omega$ for brevity. If the function f is defined by an SH expansion as $f = \sum_{j=1}^{m^2} l_j Y_j$, Equation (14) becomes

$$\sum_{j=1}^{m^2} l_j \, \langle g_k, Y_j \rangle - \sum_{i=1}^{M^2} l_i^g \, \langle g_k, g_i \rangle = 0 \quad \Leftrightarrow \quad \vec{l^g} = \mathcal{G}^{-1} \, \mathcal{C} \, \vec{l},$$

with $\mathcal{C}_{ji} = \langle g_j, Y_i \rangle$ and $\mathcal{G}_{ji} = \langle g_j, g_i \rangle$. The matrix $\mathcal{G}$ is called the *Gram matrix* of the basis. It is guaranteedly nonsingular if the basis set is linearly independent.