Eurographics Symposium on Rendering (2007)
Jan Kautz and Sumanta Pattanaik (Editors)

Precomputed Radiance Transfer for Dynamic Scenes Taking into Account Light Interreflection

Kei Iwasaki[1] and Yoshinori Dobashi[2] and Fujiichi Yoshimoto[1] and Tomoyuki Nishita[3]

[1]Wakayama University
[2]Hokkaido University [3]The University of Tokyo

Abstract

Fast rendering of dynamic scenes taking into account global illumination is one of the most challenging tasks in computer graphics. This paper proposes a new precomputed radiance transfer (PRT) method for rendering dynamic scenes of rigid objects taking into account interreflections of light between surfaces with diffuse and glossy BRDFs. To compute the interreflections of light between rigid objects, we consider the objects as secondary light sources. We represent the intensity distributions on the surface of the objects with a linear combination of basis functions. We then calculate a component of the irradiance per basis function at each vertex of the object when illuminated by the secondary light source. We call this component of the irradiance, the basis irradiance. The irradiance is represented with a linear combination of the basis irradiances, which are computed efficiently at run-time by using a PRT technique. By using the basis irradiance, the calculation of multiple-bounce interreflected light is simplified and can be evaluated very quickly. We demonstrate the real-time rendering of dynamic scenes for low-frequency lighting and rendering for all-frequency lighting at interactive frame rates.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Radiosity I.3.7 [Computer Graphics]: Animation I.3.3 [Computer Graphics]: Display Algorithm

1. Introduction

The effects of light interreflections are important for enhancing the realism of computer-generated images. Several methods have been proposed to render static scenes taking into account light interreflections in real-time. In dynamic scenes, however, the computation of the interreflection at interactive frame rates is very difficult because of the frequent changes of information relating to occlusion due to changes in the location of the objects and the light sources. However, in applications such as lighting simulations and games, the objects and the light sources are often moved interactively. Thus, the soft shadows and light interreflections in such scenes need to be computed in real-time.

There are many methods for generating images that take into account global illumination effects; for instance, Monte Carlo ray tracing [Kaj86], radiosity methods [NN85], and photon mapping [JC98]. These methods can generate high-quality photorealistic images; however, real-time rendering cannot be achieved due to the high computational cost.

Recently, precomputed radiance transfer (PRT) [SKS02, NRH03] methods have been proposed for the real-time rendering of static scenes while taking into account soft shadows and light interreflections. However, it is assumed that, when using these methods, the relative positions between all of the objects in the scene are fixed and therefore they cannot be applied to dynamic scenes. Several techniques, including precomputed shadow field methods, [ZHL*05,RWS*06, TJCN06] have been proposed for the fast computation of soft shadows in dynamic scenes. However, these methods do not consider light interreflections.

In order to compute the light interreflections between objects, we treat the objects as secondary light sources. We represent the intensity distribution of the secondary light source as a linear combination of basis functions. A component of the source radiance from the secondary light source that corresponds to each basis function, is referred to as the *basis radiance*. The source radiance from a secondary light source that has a changing intensity distribution can be represented by a linear combination of basis radiances. This makes it

Figure 1: *Rendering a dynamic scene taking into account multiple bounce interreflection of light. This scene contains four teapots with glossy BRDFs and 26 teapots with diffuse BRDFs in a room, illuminated by a square light source. Interreflected light due to yellow and purple teapots can be seen in a glossy teapot. The rendering frame rates for relighting, changing the viewpoint, changing the reflectance are about 6.0 fps. Our method can move and rotate teapots at interactive frame rates.*

possible to precompute a field that records the basis radiance at sample points around the object from the secondary light source. We call this field, the *basis radiance field*. By using the basis radiance field, we calculate a component of the irradiance per basis function, called the *basis irradiance*. The irradiance from the secondary light source is represented by a linear combination of the basis irradiances. The basis irradiance is calculated "on-the-fly" by integrating the precomputed basis radiances, the self-occlusion, and the occlusions due to other objects, which are all precomputed and therefore calculated efficiently by using a PRT technique. The interreflected light is calculated by multiplying the weighted sum of the basis irradiances by the BRDF. By incorporating the basis irradiance and the separable BRDF approximations, our method can deal with low-frequency glossy BRDFs.

The main features of our method are as follows:

- efficient calculation of multiple-bounce light interreflections in dynamic scenes, where rigid objects and light sources move,
- diffuse and glossy BRDFs,
- local light sources and distant lighting represented by an environment map,
- real-time rendering using GPU for low-frequency lighting and rendering at interactive frame rate for all-frequency lighting.

This paper is organized as follows. Section 2 reviews previous work. Section 3 describes the basic idea of our method. Section 4 explains the precomputation process and Section 5 describes the rendering process. Section 6 shows the results of the rendering process, and the conclusions to this work are brought together in Section 7.

2. Previous Work

We first review previous methods based on radiosity, and then previous methods for PRT are discussed.

Radiosity : Radiosity methods [NN85, GTGB84, CG85] were developed to create realistic images of scenes while taking into account interreflections. Cohen et al. [CCWG88] presented a progressive refinement approach. Hanrahan et al. [HSA91] proposed a hierarchical radiosity method to solve the radiosity equation rapidly. Although these methods can be used to create realistic images efficiently, real-time rendering of dynamic scenes cannot be achieved. Zatz [Zat93] proposed an alternative radiosity algorithm that represents the radiance of the surface by using polynomials. This method, however, requires shadow-masking to render shadows. Gortler et al. [GSCH93] represented the radiosity of the surface using wavelets as basis functions. Sillion proposed a radiosity method that supported complex reflectance distributions by using spherical harmonics [SAWG91]. Although these methods can be used to render realistic images, they do not deal with dynamic scenes. Drettakis and Sillion presented a method that updated the illumination interactively using the line-space hierarchy [DS97]. This method, however, does not deal with glossy surfaces.

PRT for Static Scenes : Dobashi et al. [DKYN95] proposed a method that uses spherical harmonics for computing the intensities at the surfaces of objects and which can even achieve fast rendering even when the distribution of the light sources is changed. Moreover, Dobashi et al. [DKYN96] used Fourier series to represent the intensity distributions at the surfaces of objects illuminated by light from a skylight. This method realized fast rendering when the intensity distribution of the skylight changed due to the changing the position of the sun. Ramamoorthi et al. [RH02] rendered scenes under environment illumination in real-time using spherical harmonics. Sloan et al. [SKS02] proposed the PRT technique for real-time rendering under environment illumination using the spherical harmonics as the basis functions. Several methods [KSS02, LK03, SLSS03, SLS05] have been proposed to extend the PRT method. However, these methods only considered low-frequency lighting environments and the objects in the scenes could not be moved.

Ng et al. [NRH03] used wavelet basis functions for a non-linear lighting approximation and achieved interactive rendering for diffuse surfaces. Several methods [WTL04, LSSS04, TS06, WNLH06] have been proposed to render glossy surfaces. These methods can create images under all-frequency lighting environments; however, the objects in the scenes cannot be moved.

Kristensen et al. [KMJ05] presented a PRT method for local light sources in which the lights could be moved and extra lights added. Hill et al. [SHF06] proposed an off-

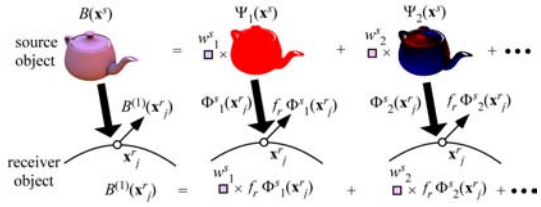


Figure 2: *Representation of intensity distribution on a source object with a linear combination of basis functions Ψ_i. Red denotes the positive value of basis function and blue denotes negative. The basis irradiance Φ is calculated for each vertex of a receiver object from each basis function Ψ_i.*

set radiance transfer to take into account indirect illumination. Kontkanen et al. [KTHS06] presented an interactive relighting method that can handle diffuse and glossy surfaces. Hasan et al. [HPB06] presented a method involving direct-to-indirect transfer for relighting complex scenes interactively. Although these methods can be used to relight a scene efficiently, the objects in the scenes still cannot be moved.

PRT for Dynamic Scenes : Sloan et al. [SKS02] presented a neighborhood transfer to render soft shadows due to moving objects or moving lights. This method, however, cannot deal with multibounce interreflections. Kautz et al. [KLA04] proposed a fast rendering method for soft shadows using hemispherical rasterization. Mei et al. [MSW04] presented a method that stores the occlusion and reflection information for the object in Spherical Radiance Transport Maps. This method can render soft shadows and glossy interreflections in dynamic scenes very quickly. However, this method cannot deal with the interreflection of light between diffuse surfaces and local light sources. Tamura et al. [TJN05] distinguished between self-shadow and shadows cast by other objects, and rapidly computed shadows cast by other objects by using occlusion maps. Zhou et al. [ZHL*05] presented a precomputed shadow field method for the interactive rendering of soft shadows in dynamic scenes. Kontkanen et al. [KL05] proposed an ambient occlusion fields to render soft shadows. Tamura et al. [TJCN06] presented adaptive shadow fields and GPU-based rendering of dynamic scenes under low-frequency lighting. Ren et al. [RWS*06] presented a spherical harmonics exponentiation technique and achieved real-time rendering of soft shadows in dynamic scenes under low-frequency lighting environments. Sun and Mukherjee [SM06] presented a generalized wavelet product integral method and rendered dynamic glossy objects interactively. Although these methods can deal with dynamic scenes, interreflections of light between surfaces could not be considered.

In summary, none of the above-mentioned methods can achieve the fast rendering of dynamic scenes while taking into account the interreflection of light. In this paper, we propose a method that can be used to realize this goal.

3. Basic Idea

This section describes the basic idea of calculating light interreflections. We first explain the concept of basis irradiance and describe the fast calculation of indirect illumination using basis irradiance.

3.1. Basis Irradiance

We calculate the interreflection of light between objects by considering each object as a secondary light source. In this paper, we define an object that is considered to be a secondary light source as a *source object*, and an object that is illuminated by a source object is referred to as a *receiver object*. For the sake of the simplicity, we will explain our method under the assumption that the surfaces of the object are diffuse and that there are no occlusions due to other objects and/or the object itself, although our method can handle glossy reflections and occlusions between objects. To calculate the interreflected light at the receiver object, the incident light from the source object that arrives at each vertex of the receiver objects has to be calculated. However, making these calculations by using the traditional approaches is time consuming, since it requires the tracing many rays from each vertex of the receiver object. To accelerate this calculation, we represent the intensity distribution on the source object by a linear combination of basis functions (see upper part of Figure 2). That is, the radiance $B(\mathbf{x}^s, \omega_{out})$ at $\mathbf{x}^s$ of the source object is represented by

$$B(\mathbf{x}^s, \omega_{out}) = \sum_{i=1}^{N} w_i^s \Psi_i(\mathbf{x}^s), \tag{1}$$

where Ψ represents the basis function, w_i^s is the weight of i-th basis function Ψ_i for the source object, and N is the number of basis functions. We generate sample points that are distributed on concentric spheres around the source object. In our implementation, concentric spheres are spaced uniformly for the simplicity. At each sample point $\mathbf{p}$, we precompute the basis radiance S_i from the source object whose intensity distribution is Ψ_i. $S_i(\mathbf{p}, \omega_{in})$ is expressed by

$$S_i(\mathbf{p}, \omega_{in}) = \Psi_i(\tilde{\mathbf{x}}^s(\mathbf{p}, \omega_{in})), \tag{2}$$

where $\tilde{\mathbf{x}}^s(\mathbf{p}, \omega_{in})$ is the intersection point between the source object and a ray from $\mathbf{p}$ in direction ω_{in} (see Figure 3). $S_i(\mathbf{p}, \omega_{in})$ is represented by spherical harmonics for low-frequency lighting and Haar wavelets for all-frequency lighting. The corresponding coefficients for spherical harmonics or Haar wavelets are stored at each sample point. A similar idea was introduced in precomputed shadow fields [ZHL*05]. The main difference in our approach is that a light source whose intensity distribution changes can be handled by updating the weights w_i^s, whereas the precomputed shadow fields cannot deal with changes such as these.

The radiance $B^{(1)}(\mathbf{x}_j^r)$ of the first bounce interreflected light at each vertex $\mathbf{x}_j^r$ of the receiver object is calculated

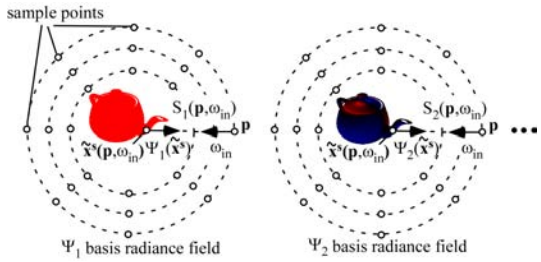


Figure 3: *Basis radiance fields. We precompute N basis radiance fields per object.*

by using the following equation:

$$B^{(1)}(\mathbf{x}_j^r, \omega_{out}) = f_r \int_\Omega L(\mathbf{x}_j^r, \omega_{in})(\mathbf{n}_j^r \cdot \omega_{in}) d\omega_{in}, \qquad (3)$$

where f_r is the diffuse reflectance of the receiver object, Ω is the hemisphere of the incoming directions, $L(\mathbf{x}_j^r, \omega_{in})$ is the incident radiance, $\mathbf{n}_j^r$ is the normal at $\mathbf{x}_j^r$, and ω_{in} is the direction of the incident. The incident radiance $L(\mathbf{x}_j^r, \omega_{in})$ is obtained by the radiance $B(\tilde{\mathbf{x}}^s, \omega_{out})$ of the source object at $\tilde{\mathbf{x}}^s$. The term $B(\tilde{\mathbf{x}}^s, \omega_{out})$ can be rewritten by using the basis radiance, so therefore we obtain $L(\mathbf{x}_j^r, \omega_{in}) = \sum_{i=1}^N w_i^s S_i(\mathbf{x}_j^r, \omega_{in})$. By substituting this into Eq.(3), $B^{(1)}(\mathbf{x}_j^r)$ can be expressed by using the basis irradiances Φ_i^s as :

$$\begin{aligned} B^{(1)}(\mathbf{x}_j^r, \omega_{out}) &= f_r \sum_{i=1}^N w_i^s \left(\int_\Omega S_i(\mathbf{x}_j^r, \omega_{in})(\mathbf{n}_j^r \cdot \omega_{in}) d\omega_{in} \right) \\ &= f_r \sum_{i=1}^N w_i^s \Phi_i^s(\mathbf{x}_j^r). \qquad (4) \end{aligned}$$

By representing the intensity distribution of the source object with basis functions, the interreflected light from the source object is calculated by multiplying the weighted sum of the basis irradiances by the diffuse reflectance. The term $(\mathbf{n}_j^r \cdot \omega_{in})$ is precomputed at each vertex $\mathbf{x}_j^r$ and represented by spherical harmonics or Haar wavelets. Therefore, the basis irradiance $\Phi_i^s(\mathbf{x}_j^r)$ is calculated very quickly by the dot product of the corresponding coefficient vectors of S_i and $(\mathbf{n}_j^r \cdot \omega_{in})$.

3.2. Fast calculation of interreflected light using basis irradiances

Figure 4 shows an overview of our rendering process using basis irradiances. Our rendering process consists of four steps as shown in Figure 4(a). The direct illumination of each object is calculated by using precomputed shadow fields [ZHL*05]. The basis irradiance of a receiver object is calculated by considering each object in the scene as a source object.

Let us now explain the calculation of indirect illumination between two objects as shown in Figure 4(b). Once the direct illumination and the basis irradiances have been computed, we calculate their weights based on the intensity distributions of object O_1 and object O_2 due to direct illumination. We then compute the first bounce interreflected light at each vertex of O_2 by considering O_1 as a source object. This calculation is evaluated efficiently by substituting the weights of O_1 in Eq.(4). Similarly, the first bounce interreflected light of O_1 is calculated by considering O_2 as a source object. Then the intensity distributions on O_1 and O_2 due to first bounce interreflected light are computed. We update the weights for O_1 and O_2 by using these intensity distributions of the first bounce interreflected light. The second-bounce interreflected light for each object is calculated by using the updated weights. Multiple-bounce interreflected light can be calculated efficiently by repeating these steps. That is, the indirect illumination can be calculated very quickly by updating the weights and by a linear combination of the basis irradiances.

In the rendering process, the computational cost to obtain the basis irradiance is relatively expensive compared to the costs of other steps. If there are N_{obj} objects in a scene, each object receives basis irradiances from $(N_{obj} - 1)$ other objects. Therefore, $(N_{obj} - 1) \times N$ basis irradiances are calculated for each vertex of each object. We can accelerate the calculation of the basis irradiance. The basis irradiance depends on the relative positions of each vertex of the receiver and the source objects, and the occlusions that occur between them. Therefore, the basis irradiance is only updated in those cases where the relative positions or the occlusion information have been changed. Moreover, we can further accelerate the calculation of the basis irradiance by omitting those instances of basis irradiance that make only a small contribution. The acceleration method is described in Section 5.4.

To represent the intensity distributions on the source objects, we use basis functions that are calculated by Principal Component Analysis (PCA) for various illumination distributions. We call these basis functions *PCA basis functions*.

4. Precomputation of Basis Radiance Fields

We will now describe the calculation method for interreflected light for glossy surfaces taking into account occlusions due to objects. To calculate the source radiance from a source object, the intensity distribution of the source object must be calculated. Then the intensity distribution is approximated by using basis functions. Therefore, we first describe the calculation of the intensity distribution of the source object.

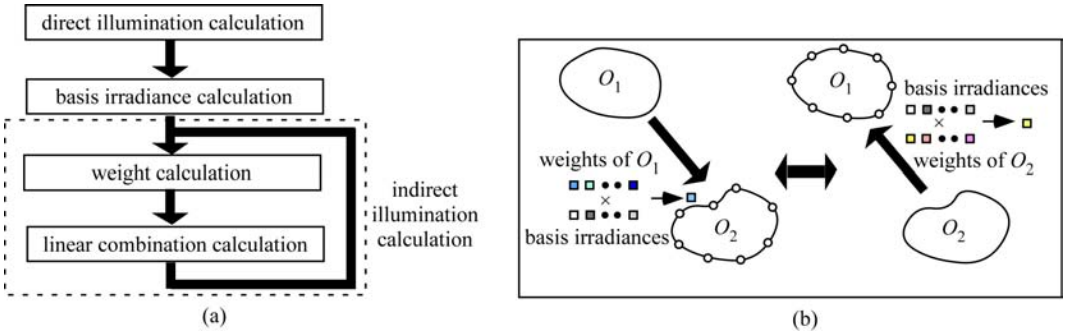


Figure 4: *(a) Overview of the four steps of our rendering method. We first calculate the direct illumination of each object. Next, we calculate the basis irradiance at each vertex of the receiver object per source object and per basis function. Then we calculate weights for each object and basis function. Interreflected light is calculated by the product of weighted sum of basis irradiance and BRDF. (b) once the basis irradiance has been calculated, interreflected light is calculated by repeating the update of weights and the calculation of the weighted sum of basis irradiance using updated weights.*

4.1. Dimension reduction of basis functions using BRDF factorization

The outgoing radiance at $\mathbf{x}_j^s$ of a source object in the direction ω_{out} is calculated by

$$B(\mathbf{x}_j^s,\omega_{out})=\int_\Omega L(\mathbf{x}_j^s,\omega_{in})V(\mathbf{x}_j^s,\omega_{in})f_r(\omega_{in},\omega_{out})(\omega_{in}\cdot\mathbf{n}_j^s)d\omega_{in}, \quad (5)$$

where L is the lighting, V is the visibility function, and f_r is the BRDF. We calculate L and V by using the precomputed shadow field [ZHL*05]. To calculate the light interreflection, we represent $B(\mathbf{x}_j^s,\omega_{out})$ by a linear combination of the PCA basis function $\Psi_i(\mathbf{x}_j^s,\omega_{out})$ as $B(\mathbf{x}_j^s,\omega_{out}) = \sum_{i=1}^N w_i^s\Psi_i(\mathbf{x}_j^s,\omega_{out})$. The intensity distribution function of the source object is four-dimensional (two dimensions for position and two for outgoing direction) and therefore a large number of basis functions are required to represent it. In terms of rendering speed and memory consumption, we would like to represent the intensity distribution by using a smaller number of basis functions.

For this purpose, we need to reduce the dimensionality of the basis functions. In Eq. (5), the outgoing direction ω_{out} is only involved with the BRDF f_r. Therefore, we approximate the BRDF f_r by using separable decomposition [KM99] as,

$$f_r(\omega_{in},\omega_{out})\approx\sum_{k=1}^K g_k(\omega_{in})h_k(\omega_{out}), \quad (6)$$

where K is the number of approximation terms, and g_k and h_k are the k-th functions depending on the incident and outgoing directions, respectively. For diffuse BRDF, $K=1$ and $h_k(\omega_{out})=1$. By using the separable approximation, $B(\mathbf{x}_j,\omega_{out})$ is calculated by

$$\int_\Omega L(\mathbf{x}_j^s,\omega_{in})V(\mathbf{x}_j^s,\omega_{in})\left(\sum_{k=1}^K g_k(\omega_{in})h_k(\omega_{out})\right)(\omega_{in}\cdot\mathbf{n}_j^s)d\omega_{in}$$
$$=\sum_{k=1}^K h_k(\omega_{out})\int_\Omega L(\mathbf{x}_j^s,\omega_{in})V(\mathbf{x}_j^s,\omega_{in})g_k(\omega_{in})(\omega_{in}\cdot\mathbf{n}_j^s)d\omega_{in}. \quad (7)$$

We define the integral on the right side of Eq. (7) as $L_k(\mathbf{x}_j^s)$.

$$L_k(\mathbf{x}_j^s)=\int_\Omega L(\mathbf{x}_j^s,\omega_{in})V(\mathbf{x}_j^s,\omega_{in})g_k(\omega_{in})(\omega_{in}\cdot\mathbf{n}_j^s)d\omega_{in}. \quad (8)$$

In dynamic scenes where objects and light sources move, the incident lighting L and the visibility V change, and therefore L_k changes. On the other hand, the function $h_k(\omega_{out})$ is independent of the changes of other objects and the positions of light sources. Therefore, we can precompute $h_k(\omega_{out})$ at each sample point (see Figure 3) around the source object. Consequently, $L_k(\mathbf{x}_j^s)$ can be represented by a linear combination of PCA basis functions, resulting in a reduction in the dimensionality of the basis functions. $L_k(\mathbf{x}_j^s)$ and $B(\mathbf{x}_j^s,\omega_{out})$ are expressed by:

$$\begin{aligned} L_k(\mathbf{x}_j^s) &= \sum_{i=1}^N w_{(k,i)}^s\Psi_i(\mathbf{x}_j^s), \\ B(\mathbf{x}_j^s,\omega_{out}) &= \sum_{k=1}^K\sum_{i=1}^N w_{(k,i)}^s h_k(\omega_{out})\Psi_i(\mathbf{x}_j^s). \end{aligned} \quad (9)$$

4.2. Calculation of PCA basis functions

To calculate the PCA basis functions of the source object, our method uses point light sources to illuminate the source object. N_{light} point light sources are distributed on the bounding sphere of the source object as shown in Figure 5. In our experiments, N_{light} is set to 256. We calculate the reflected light $B_\eta(\mathbf{x}_j^s)$ at vertex $\mathbf{x}_j^s$ for each point light source P_η. We represent the intensity distribution on the source object as a column vector $\mathbf{B}_\eta$ whose j-th element is $B_\eta(\mathbf{x}_j^s)$. We calculate the mean vector $\mathbf{B}_0$ of vectors $\mathbf{B}_1,\mathbf{B}_2,\cdots,\mathbf{B}_{N_{light}}$. We then calculate a $(N_v\times N_{light})$ matrix $\mathbf{C}=\{\mathbf{B}_1-\mathbf{B}_0,\mathbf{B}_2-\mathbf{B}_0,\cdots,\mathbf{B}_{N_{light}}-\mathbf{B}_0\}$, where N_v is the number of vertices of the source object. We calculate the Singular Value Decomposition of $\mathbf{C}$ as $\mathbf{C}=\mathbf{USV}^T\approx\sum_{\mu=1}^{N-1}\sigma_\mu\mathbf{u}_\mu\mathbf{v}_\mu^T$, where T means the transpose of the matrix or the vector, the

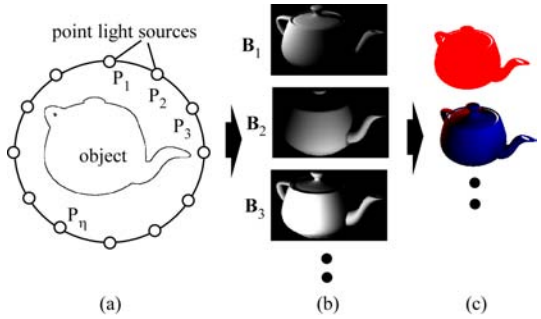


Figure 5: *Calculation of PCA basis functions. (a) We set point light sources around the source object. (b) Reflected light at each vertex illuminated by each point light source is calculated. (c) We calculate principal component analysis for the reflected light.*

singular values σ_μ are diagonal elements of $\mathbf{S}$, and $\mathbf{u}_\mu$, $\mathbf{v}_\mu$ are column vectors of $\mathbf{U}$ and $\mathbf{V}$. The first $(N-1)$ vectors $\mathbf{u}_\mu$ are the PCA vectors. By using $(N-1)$ PCA vectors $\mathbf{u}_\mu$ and the mean vector $\mathbf{B}_0$, the reflected light at $\mathbf{x}_j^s$ is represented by $B_\eta(\mathbf{x}_j^s) = B_0(\mathbf{x}_j^s) + \sum_{\mu=1}^{N-1} w_\mu u^\mu(\mathbf{x}_j^s) = \sum_{\mu=0}^{N-1} w_\mu u^\mu(\mathbf{x}_j^s)$, where $u^\mu(\mathbf{x}_j^s)$ is the j-th element of $\mathbf{u}^\mu$. We define the values $u^\mu(\mathbf{x}_j^s)$ as PCA basis functions $\Psi_\mu(\mathbf{x}_j^s)$.

4.3. Precomputation of basis radiance

By using separable BRDF approximations and basis functions, the outgoing radiance $B(\mathbf{x}_j^s, \omega_{out})$ is represented by a linear combination of $h_k(\omega_{out})\Psi_i(\mathbf{x}_j^s)$ as shown in Eq.(9). Therefore, we can precompute the basis radiance at each sample point $\mathbf{p}$. The basis radiance $S_{(k,i)}(\mathbf{p}, \omega_{in})$ is equal to the outgoing radiance in the direction ω_{out} at $\tilde{\mathbf{x}}^s(\mathbf{p}, \omega_{in})$, that is the intersection point between the ray from $\mathbf{p}$ in direction ω_{in}. Since $\omega_{out} = -\omega_{in}$, the basis radiance $S_{(k,i)}(\mathbf{p}, \omega_{in})$ is expressed by $S_{(k,i)}(\mathbf{p}, \omega_{in}) = h(-\omega_{in})\Psi_i(\tilde{\mathbf{x}}^s(\mathbf{p}, \omega_{in}))$. We calculate $\Psi_i(\tilde{\mathbf{x}}^s(\mathbf{p}, \omega_{in}))$ by interpolating the basis functions of the vertices of the face intersecting the ray at $\tilde{\mathbf{x}}^s$. We project $S_{(k,i)}(\mathbf{p}, \omega_{in})$ onto spherical harmonics for low-frequency application or as a Haar wavelet for all-frequency application, and we then store the corresponding coefficients.

5. Rendering

In this section, we describe the details of basis irradiance calculation, the weight calculation, and the calculation of interreflected light for glossy surfaces.

5.1. Calculation of basis irradiance

We first describe the representation of the outgoing radiance of the interreflected light with the basis irradiance and the separable BRDFs. The radiance $B^{(1)}$ at $\mathbf{x}_j^r$ of the first bounce interreflected light is calculated from the following equation:

$$B^{(1)}(\mathbf{x}_j^r, \omega_{out}) = \int_\Omega L(\mathbf{x}_j^r, \omega_{in}) f_r(\omega_{in}, \omega_{out})(\mathbf{n}_j^r \cdot \omega_{in}) d\omega_{in}, \tag{10}$$

where $L(\mathbf{x}_j^r, \omega_{in})$ is the incident radiance. The BRDF $f_r(\omega_{in}, \omega_{out})$ of the receiver object is also separated as $f_r(\omega_{in}, \omega_{out}) = \sum_{k'=1}^{K} g_{k'}(\omega_{in}) h_{k'}(\omega_{out})$, where K is number of approximation terms. By substituting these in Eq.(10), Eq.(10) can be rewritten as:

$$B^{(1)}(\mathbf{x}_j^r, \omega_{out}) = \sum_{k'=1}^{K} h_{k'}(\omega_{out}) \int_\Omega L(\mathbf{x}_j^r, \omega_{in}) g_{k'}(\omega_{in})(\mathbf{n}_j^r \cdot \omega_{in}) d\omega_{in}. \tag{11}$$

The incident radiance L is calculated by using the basis radiance as $L(\mathbf{x}_j^r, \omega_{in}) = \sum_{k=1}^{K}\sum_{i=1}^{N} w_{(k,i)} S_{(k,i)}(\mathbf{x}_j^r, \omega_{in}) V(\mathbf{x}_j^r, \omega_{in})$, where V is a visibility function that includes the self-occlusion and occlusions due to other objects between the source and the receiver objects. By substituting this in Eq. (11), Eq. (11) is rewritten as:

$$B^{(1)}(\mathbf{x}_j^r, \omega_{out}) = \sum_{k'=1}^{K}\sum_{k=1}^{K}\sum_{i=1}^{N} w_{(k,i)}^s h_{k'}(\omega_{out}) \Phi_{(k',k,i)}^s(\mathbf{x}_j^r) \tag{12}$$

where $\Phi_{(k',k,i)}^s(\mathbf{x}_j^r)$ is expressed by:

$$\Phi_{(k',k,i)}^s(\mathbf{x}_j^r) = \int_\Omega S_{(k,i)}(\mathbf{x}_j^r, \omega_{in}) V(\mathbf{x}_j^r, \omega_{in}) g_{k'}(\omega_{in})(\mathbf{n}_j^r \cdot \omega_{in}) d\omega_{in}. \tag{13}$$

$\Phi_{(k',k,i)}^s(\mathbf{x}_j^r)$ is referred to as the basis irradiance for glossy surfaces.

The basis irradiance $\Phi_{(k',k,i)}^s$ is computed on-the-fly as follows. Basis radiance $S_{(k,i)}(\mathbf{x}_j^r, \omega_{in})$ is interpolated by the basis radiance at the nearest sample points of $\mathbf{x}_j^r$. As described in Section 4.3, the basis radiance is represented by a vector of coefficients of spherical harmonics for low-frequency lighting or by Haar wavelets for all-frequency lighting. The visibility function $V(\mathbf{x}_j^r, \omega_{in})$ is calculated by using the precomputed shadow field [ZHL*05] and is represented by a coefficient vector. $g_{k'}(\omega_{in})(\mathbf{n}_j^r \cdot \omega_{in})$ is also precomputed at each vertex $\mathbf{x}_j^r$ and represented by a coefficient vector. Since these functions are expressed in orthonormal bases (spherical harmonics or Haar wavelets), the integration of these functions is reduced to the products of the coefficient vectors and can be calculated efficiently. To handle rotations of objects for low-frequency lighting, these functions are rotated by using spherical harmonics rotations [KSS02].

5.2. Calculation of weights

We calculate weight $w_{(k,i)}^s$ to minimize the sum of squares differences between $L_k(\mathbf{x}_j^s)$ and a linear combination of N basis functions $\sum_{i=1}^{N} w_{(k,i)}^s \Psi_i(\mathbf{x}_j^s)$ in all the

vertices of the source object. We define a vector $\mathbf{w}_k^s = (w_{(k,1)}^s, w_{(k,2)}^s, \cdots, w_{(k,i)}^s, \cdots, w_{(k,N)}^s)^T$ and the sum of square differences $F(\mathbf{w}_k^s)$ as the following equation,

$$F(\mathbf{w}_k^s) = \sum_{j=1}^{N_v} \left[L_k(\mathbf{x}_j^s) - \sum_{i=1}^{N} w_{(k,i)}^s \Psi_i(\mathbf{x}_j^s) \right]^2, \qquad (14)$$

where N_v is the number of vertices of the source object. We calculate the weight vector $\mathbf{w}_k$ to minimize F by the method of least squares. That is, we calculate the simultaneous linear equations $\mathbf{A}\mathbf{w}_k = \mathbf{b}_k$, where $\mathbf{A}$ is a $(N \times N)$ matrix of the coefficients of the equations, and element $a_{(m,n)}$ of $\mathbf{A}$ is calculated by $\sum_{j=1}^{N_v} \Psi_m(\mathbf{x}_j^s)\Psi_n(\mathbf{x}_j^s)$. Element b_m of $\mathbf{b}_k$ is calculated from $\sum_{j=1}^{N_v} \Psi_m(\mathbf{x}_j^s)L_k(\mathbf{x}_j^s)$. Since matrix $\mathbf{A}$ depends only on basis functions, inverse matrix $\mathbf{A}^{-1}$ can be calculated in advance. Therefore, the weight vector $\mathbf{w}_k^s$ can be calculated efficiently as the product of a precomputed matrix $\mathbf{A}^{-1}$ and $\mathbf{b}_k$ in the rendering process.

5.3. Calculation of interreflected light

As shown in Eq. (12), the calculation of the interreflected light is simplyfied considerably by using the basis irradiance technique and can therefore be calculated very quickly. First bounce interreflected light is calculated by summing $B^{(1)}$ in Eq. (12) from all objects in the scene. Then we update weights by using the intensity distributions on surfaces of all of the objects. Multiple-bounce interreflected light can be calculated efficiently by repeatedly updating the weights and then running the calculation in Eq. (12).

5.4. Acceleration of the basis irradiance calculation

We can accelerate the rendering process by omitting the calculations of the basis irradiance that will only make a small contribution for each object. If the source object is far from the receiver object, the basis radiance and the basis irradiance are small. Therefore, we do not calculate the basis irradiance in cases like this that only make a small contribution. We determine the basis irradiance that is to be calculated as follows. Let $\mathbf{x}_c^s$ and $\mathbf{x}_c^r$ be the centers of the bounding spheres of the source and the receiver objects, respectively and ω_c be a unit vector of $\mathbf{x}_c^s - \mathbf{x}_c^r$. We then calculate $I_c = \int_\Omega S_{(k,i)}(\mathbf{x}_c^r, \omega_{in}) g_{k'}(\omega_{in})(\omega_c \cdot \omega_{in}) d\omega_{in}$ for each term k', k, and i. I_c indicates the reflected light at the plane perpendicular to direction ω_c. If I_c is less than a threshold value, then the interreflected light at each vertex is less than the threshold. Therefore, we only calculate the interreflected light from the $S_{(k,i)}$ in case where I_c is greater than a threshold value that is specified by the user.

5.5. Implementation on GPU

We have implemented our rendering method for low-frequency lighting on a GPU. The calculations for the direct illumination, the basis irradiances and the linear combination of basis irradiance are performed on the GPU. The weight calculation is performed on the CPU, since it requires the products of matrices and vectors, which is quite expensive to perform on a GPU. We prepare a floating point texture for each object, and each pixel of the texture corresponds to each vertex of the object. Then the direct illumination, the basis irradiance and the linear combination of basis irradiance are calculated using a fragment program. The rendering speed using the GPU is about 6 to 12 times faster than that using the CPU in our experiments.

6. Results and Discussions

We have tested our algorithm on a laptop PC equipped with Core2Duo 2.16GHz with 2GB memory and GeForce Go 7900 GTX. Figure 6 shows a room scene containing 12 objects illuminated by one square light source. The rendering frame rate of Figure 6 is 5.0-19.0fps (frames per second). The computational times for the direct illumination and the basis irradiances are 0.057 and 0.11 seconds, respectively. The total computational time for the indirect illumination calculation of three bounce interreflected light is 0.033 seconds.

The number of vertices in this figure is 50K. The data size of the precomputed basis radiance fields is 66.3MB. Figure 7 shows a scene containing four teapots with glossy BRDFs, 26 teapots with diffuse BRDFs and in a room under square light source. The rendering frame rate of Figure 7 for changing the light source, the viewpoint, and the diffuse reflectance is 6.0fps, and the rendering frame rate for translating 30 teapots is 0.6 fps. As shown in this figure, interreflected light due to cyan and green teapots can be seen in the glossy teapots. The number of vertices in this figure is 122K. The data size of precomputed basis radiance fields is 37.7MB. Please refer to the accompanied animation.

Figure 8 shows a model of the Buddha illuminated by all-frequency lighting. Figure 8(a) is rendered by taking into account the interreflection of light, and Figure 8(b) is rendered taking into account only direct illumination. The model Buddha becomes reddish due to the light reflected from the red floor. The rendering frame rate in Figure 8 is 0.1fps. The number of vertices in this figure is 14K. The data size of precomputed basis radiance fields is 56.0MB. Figure 8 is rendered at only interactive rates since the computational costs for the products of coefficient vectors for Haar wavelets are relatively expensive.

The number N of PCA basis functions used in these figures is 16 and the number K of BRDF approximation terms is set to 4, so that it is the same as it is for the previous method [WTL04]. The number of sample points that are used to record the basis radiance fields is 24,576 (16 concentric spheres and $6 \times 16 \times 16$ sample points are uniformly distributed on each concentric sphere). We calculate bounced interreflected light three times in these figures.

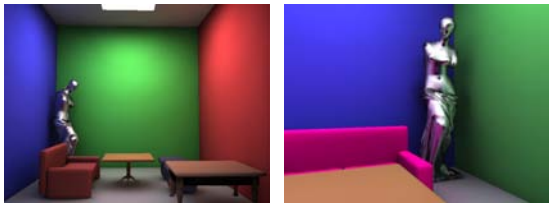


Figure 6: *Room scene containing 11 objects with diffuse BRDFs and a statue of venus with glossy phong BRDF. Interreflected light due to sofa and green wall can be seen in the glossy venus statue.*

Figure 7: *Teapots scene containing four glossy teapots and 26 teapots with diffuse BRDF in a room.*

(a) (b)

Figure 8: *A model of the Buddha with glossy BRDF under all-frequency lighting. (a) direct and indirect illumination (b) direct illumination only.*

(a) (b) (c)

Figure 9: *Comparison between images rendered by using (a) our method and (b) Monte Carlo method. (c) shows the difference between (a) and (b).*

Figure 9 shows a comparison between (a) our method and (b) a Monte Carlo-based method for calculating indirect illumination. Figure 9(c) shows the difference between (a) and (b). The difference is normalized so that white color indicates the 10 % difference, where 100 % means that the difference is 255. The root-mean-square (RMS) error of the bunny model is 0.054. As shown in this figure, the image rendered by using our method is almost visually indistinguishable from an image rendered by using the Monte Carlo method for low-frequency lighting.

6.1. Discussions

The choice of the basis functions to represent the intensity distribution on the source object influences the rendering quality, the speeds, and the memory consumption for the basis radiance field. The intensity distribution can be accurately approximated by increasing the number of terms for the basis functions. However, the memory required for the basis radiance field increases and the rendering time increases in proportion to the number of terms that are used for the basis functions. Therefore, we now discuss the choices that we have for picking basis functions to represent the intensity distribution on the source object.

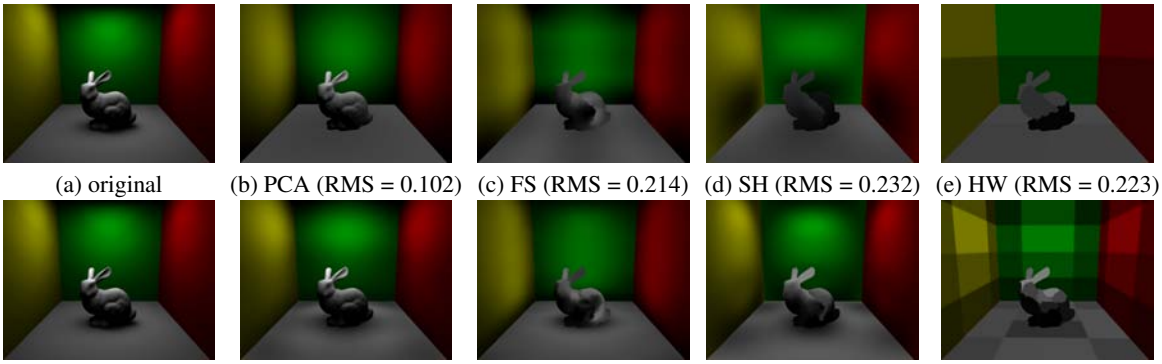
Many methods have been developed to represent a function (e.g. intensity distribution, BRDF, transfer functions) using basis functions. Fourier cosine series (FS) [DKYN96], radial basis functions (RBF) [TS06], spherical harmonics (SH) [SKS02], and Haar wavelets (HW) [NRH04] are frequently used to approximate a function. As shown in Figure 4, weights are calculated in real-time for the indirect illumination calculation. Therefore, we omit RBF since the cost of weight calculation for RBF is quite expensive.

The intensity distributions on the object surface can be parameterized in a 2D domain, since the object surface is parameterized in a 2D domain. This means that the basis functions of FS, SH, HW are also parameterized in a 2D domain. We used a stretch-minimizing parameterization [YBS05]. We have experimented with PCA basis functions, FS, SH, and HW by representing the intensity distributions due to the direct illumination of the square light source in Figure 10. The images in the upper row in Figure 10(b) to (e) are rendered by approximating the intensity distributions with 4-term basis functions, while the images in the lower row from Figure 10(g) to (j), are rendered by using 16-term basis functions. As shown in this figure, PCA basis functions represent the intensity distributions very well.

To calculate the direct and indirect illumination, occlusions due to other objects must also be calculated. To take into account occlusions of other objects, we have to sort the occluding objects in order of increasing distance. In our implementation on the GPU, we sort the objects by using the distance between the centers of the bounding spheres of the objects. We calculate the radiances and the basis irradiances of the vertices of each object simultaneously on the GPU by using this sorting results. However, the sorting results may not be applicable to some vertices if the objects are too close to each other. This problem can be solved by using the previous method [RWS*06] since the order of the occlusion does not matter when using the spherical harmonic exponentiation, but this remains to be considered for future work.

7. Conclusion and Future Work

We have proposed a real-time rendering method for dynamic scenes that takes into account interreflections of light by us-



(a) original (b) PCA (RMS = 0.102) (c) FS (RMS = 0.214) (d) SH (RMS = 0.232) (e) HW (RMS = 0.223)

(f) original (g) PCA (RMS = 0.064) (h) FS (RMS = 0.212) (i) SH (RMS = 0.185) (j) HW (RMS = 0.183)

Figure 10: *Comparisons of basis functions that represent the intensity distributions due to direct illumination. The upper row images from (b) to (e) are rendered by using 4 term basis functions and the lower row images from (g) to (j) are rendered by using 16 term basis functions. Root mean square (RMS) error of the bunny model is reported.*

ing a PRT technique. By representing the intensity distributions on the surfaces of the objects with a linear combination of PCA basis functions, we can immediately calculate the incident radiance from an object surface on which the intensity distribution changes. Then we calculate the basis irradiance, which is a component of the irradiance corresponding to each PCA basis function on-the-fly by using a PRT technique. By using the basis irradiance, the interreflected light is calculated by updating the weights corresponding to the PCA basis functions and by calculating the weighted-sum of the basis irradiances. Our method can deal with glossy surfaces by using separable approximations.

In future work, we would like to implement the calculation of direct and indirect illumination for all-frequency lighting on the GPU. Moreover, we would like to deal with textures to enhance the realism of the technique.

Acknowledgements

The authors would like to thank Prof. Nelson Max for valuable comments and suggestions. This research has been supported by the Kayamori Foundation of Informational Science Advancement.

References

[CCWG88] COHEN M., CHEN S., WALLACE J., GREENBERG D.: A progressive refinement approach to fast radiosity image generation. *Computer Graphics 22*, 4 (1988), 75–84.

[CG85] COHEN M., GREENBERG D.: A radiosity solution for complex environment. *Computer Graphics 19*, 3 (1985), 31–40.

[DKYN95] DOBASHI Y., KANEDA K., YAMASHITA H., NISHITA T.: A quick rendering method using basis functions for interactive lighting design. *Computer Graphics Forum 14*, 3 (1995), 229–240.

[DKYN96] DOBASHI Y., KANEDA K., YAMASHITA H., NISHITA T.: Method for calculation of sky light luminance aiming at an interactive architectual design. *Computer Graphics Forum 15*, 3 (1996), 112–118.

[DS97] DRETTAKIS G., SILLION F. X.: Interactive update of global illumination using a line-space hierarchy. In *SIGGRAPH'97* (1997), pp. 57–64.

[GSCH93] GORTLER S., SCHRODER P., COHEN M., HANRAHAN P.: Wavelet radiosity. In *Proc. SIGGRAPH 1993* (1993), pp. 221–230.

[GTGB84] GORAL C. M., TORRANCE K. E., GREENBERG D. P., BATTAILE B.: Modeling the interaction of light between diffuse surfaces. In *Proceedings of SIGGRAPH 1984* (1984), pp. 213–222.

[HPB06] HASAN M., PELLACINI F., BALA K.: Direct-to-indirect transfer for cinematic relighting. *ACM Transaction on Graphics 25*, 3 (2006), 1089–1097.

[HSA91] HANRAHAN P., SALZMAN D., AUPPERLE L.: A rapid hierarchical radiosity algorithm. In *Proc. SIGGRAPH 1991* (1991), pp. 197–206.

[JC98] JENSEN H., CHRISTENSEN P.: Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. SIGGRAPH 1998* (1998), pp. 311–320.

[Kaj86] KAJIYA J.: The rendering equation. *Computer Graphics 20*, 4 (1986), 143–150.

[KL05] KONTKANEN J., LAINE S.: Ambient occlusion fields. In *Proc. of Symposium on Interactive 3D Graphics and Games* (2005), pp. 41–48.

[KLA04] KAUTZ J., LEHTINEN J., ALIA T.: Hemispherical rasterization for self-shadowing of dynamic objects. In *Proc. Eurographics Workshop on Rendering 2004* (2004), pp. 179–184.

[KM99] KAUTZ J., MCCOOL M.: Interactive rendering

with arbitrary brdfs using separable approximation. In *Proc. Eurographics Workshop on Rendering 1999* (1999), pp. 281–292.

[KMJ05] KRISTENSEN A., MOLLER T., JENSEN H.: Precomputed local radiance transfer for real-time lighting design. *ACM Transactions on Graphics 24*, 3 (2005), 1208–1215.

[KSS02] KAUTZ J., SLOAN P., SNYDER J.: Fast, artitrary BRDF shading for low-frequency lighting using spherical harmonics. In *Proc. Eurographics Workshop on Rendering 2002* (2002), pp. 301–308.

[KTHS06] KONTKANEN J., TURQUIN E., HOLZSCHUCH N., SILLION F.: Wavelet radiance transport for interactive indirect lighting. In *Proc. of Eurographics Symposium on Rendering* (2006), pp. 161–171.

[LK03] LEHTINEN J., KAUTZ J.: Matrix radiance transfer. In *Proc. Symposium on Interactive 3D Graphics 2003* (2003), pp. 59–64.

[LSSS04] LIU X., SLOAN P., SHUM H., SNYDER J.: All-frequency precomputed radiance transfer for glossy objects. In *Proc. Eurographics Symposium on Rendering 2004* (2004), pp. 337–344.

[MSW04] MEI C., SHI J., WU F.: Rendering with spherical radiance transport maps. *Computer Graphics Forum 23*, 3 (2004), 281–290.

[NN85] NISHITA T., NAKAMAE E.: Continuous tone representation of three dimensional objects taking into account of shadows and interreflection. *Computer Graphics 19*, 3 (1985), 23–30.

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics 22*, 3 (2003), 376–381.

[NRH04] NG R., RAMAMOORTHI R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. *ACM Transactions on Graphics 23*, 3 (2004), 477–487.

[RH02] RAMAMOORTHI R., HANRAHAN P.: Frequency space environment map rendering. *ACM Transactions on Graphics 21*, 3 (2002), 517–526.

[RWS*06] REN Z., WANG R., SNYDER J., ZHOU K., LIU X., SUN B., SLOAN P., BAO H., PENG Q., GUO B.: Real-time soft shadows in dynamic scenes using spherical harmonics exponentiation. *ACM Transaction on Graphics 25*, 3 (2006), 977–986.

[SAWG91] SILLION F. X., ARVO J. R., WESTIN S. H., GREENBERG D. P.: A global illumination solution for general reflectance distributions. In *SIGGRAPH'91* (1991), pp. 187–196.

[SHF06] SUNSHINE-HILL B., FALOUTSOS P.: Photorealistic lighting with offset radiance transfer mapping. In *Proc. of Symposium on Interactive 3D Graphics and Games 2007* (2006), pp. 15–21.

[SKS02] SLOAN P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic scenes. *ACM Transactions on Graphics 21*, 3 (2002), 527–536.

[SLS05] SLOAN P., LUNA B., SNYDER J.: Local, deformable precomputed radiance transfer. *ACM Transactions on Graphics 24*, 3 (2005), 1216–1224.

[SLSS03] SLOAN P., LIU X., SHUM H., SNYDER J.: Bi-scale radiance transfer. *ACM Transactions on Graphics 22*, 3 (2003), 370–375.

[SM06] SUN W., MUKHERJEE A.: Generalized wavelet product integral for rendering dynamic glossy objects. *ACM Transaction on Graphics 25*, 3 (2006), 955–966.

[TJCN06] TAMURA N., JOHAN H., CHEN B., NISHITA T.: A practical and fast rendering algorithm for dynamic scenes using adaptive shadow fields. *The Visual Computer 22*, 9-11 (2006), 702–712.

[TJN05] TAMURA N., JOHAN H., NISHITA T.: Deffered shadowing for real-time rendering of dynamic scenes under environment illumination. *Computer Animation and Virtual World 16*, 3-4 (2005), 475–486.

[TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Transaction on Graphics 25*, 3 (2006), 967–976.

[WNLH06] WANG R., NG R., LUEBKE D., HUMPHREYS G.: Efficient wavelet rotation for environment map rendering. In *Proc. of Eurographics Symposium on Rendering 2006* (2006), pp. 173–182.

[WTL04] WANG R., TRAN J., LUEBKE D.: All-frequency relighting of non-diffuse objects using separable BRDF approximation. In *Proc. Eurographics Symposium on Rendering 2004* (2004), pp. 345–354.

[YBS05] YOSHIZAWA S., BALEYAEV A. G., SEIDEL H.-P.: A moving mesh approach to stretch-minimizing mesh parameterization. *International Journal of Shape Modeling 11*, 1 (2005), 25–42.

[Zat93] ZATZ H.: Galerkin radiosity: A higher order solution method for global illumination. In *Proc. SIGGRAPH 1993* (1993), pp. 213–220.

[ZHL*05] ZHOU K., HU Y., LIU S., GUO B., SHUM H.: Precomputed shadow fields for dynamic scenes. *ACM Transactions on Graphics 24*, 3 (2005), 1196–1201.