

# Adaptive Precomputation of Glossy Interreflections

*Jurgen Laurijssen*

*Philip Dutré*

*Report CW 548, June 2009*



Katholieke Universiteit Leuven  
Department of Computer Science

Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

# Adaptive Precomputation of Glossy Interreflections

*Jurgen Laurijssen*  
*Philip Dutré*

*Report CW 548, June 2009*

Department of Computer Science, K.U.Leuven

## **Abstract**

Precomputed radiance transfer methods compute a global light transport model the transfer function to enable real-time rendering of scenes under dynamic distant illumination for arbitrary view-points. Usually this transfer function is represented for directions and surface points that are defined a priori. In this paper, we present a precomputed radiance transfer technique, which adaptively chooses directions and surface points during precomputation based on heuristics that are easy to compute. We also present an efficient method for estimating our transfer function model. Both contributions combined enable us to demonstrate real-time renderings with accurate glossy interreflections.

# Adaptive Precomputation of Glossy Interreflections

---

## Abstract

*Precomputed radiance transfer methods compute a global light transport model—the transfer function—to enable real-time rendering of scenes under dynamic distant illumination for arbitrary viewpoints. Usually this transfer function is represented for directions and surface points that are defined a priori. In this paper, we present a precomputed radiance transfer technique, which adaptively chooses directions and surface points during pre-computation based on heuristics that are easy to compute. We also present an efficient method for estimating our transfer function model. Both contributions combined enable us to demonstrate real-time renderings with accurate glossy interreflections.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture;

---

## 1. Introduction

Light interacts with object surfaces in many diverse ways, creating important visual effects such as caustics or interreflections. Based on physical principles for simulating such light transport in a scene accurately, global illumination methods are capable of generating photorealistic imagery, making them a valuable tool for high-quality visualization applications. However, due to their high computational demands, these algorithms are rarely used in real-time applications. On the other hand, precomputed radiance transfer (PRT) techniques model and precompute global light transport explicitly. Based on this precomputed model, called the transfer function, PRT methods are capable of rendering scenes under dynamic lighting conditions and arbitrary viewpoints interactively.

The transfer function is usually sampled at every vertex of the scene and discretized by projection onto a fixed spherical basis. Thus the complexity of the light transport that can be described with the precomputed model depends heavily on mesh resolution and on the number of spherical basis functions. By choosing these aspects a priori, all-frequency global light transport such as glossy interreflections cannot be reproduced accurately with current PRT methods.

In this paper we present a method for precomputing the light transport model iteratively. We refine the initial mesh and add basis functions to the spherical basis during each iteration, based on previously obtained results. We use the von Mises-Fisher distribution to represent the transfer function because its parameters can be estimated very efficiently.

As a result we can render accurate glossy interreflections, as illustrated in Figure 1.

## 2. Previous work

Physically based rendering techniques have been the subject of graphics research for many years. A good overview of many of the classic algorithms can be found in [DBB06]. Precomputed radiance transfer methods [SKS02, KSS02] model the relationship between light originating at the distant environment and radiance reflected from a virtual scene, taking into account full global light transport. Efficient storage and evaluation of this relationship—expressed by the transfer function—allows real-time visualization of scenes with dynamic lighting for arbitrary viewpoints. Compression techniques based on principal component analysis [SHHS03, LK03] are used to represent the transfer function with a relatively compact dataset. These techniques are restricted to low-frequency phenomena due to the use of spherical harmonics as the spherical basis for representing the transfer function. Much research has been devoted to lifting this limitation. Ng *et al.* [NRH03] choose wavelets as a compact, all-frequency basis. Compression is performed with nonlinear approximation [DeV98]. Their technique can visualize sharp shadows and, by fixing the camera position, also glossy materials. By using separable BRDF approximation [KM99], even this limitation can be lifted [WTL04, LSS04, WTL06, TS06]. The BRDF is represented as the sum of products of purely light-dependent and purely view-dependent terms. However Mahajan *et al.* [MTR08] formally prove that only smoothly varying high-

lights can be visualized in this manner due to the limited number of terms in this sum.

An alternative for solving the all-frequency problem is to model and store components of the transfer function separately with Haar wavelets. Efficient algorithms have been developed to evaluate Haar wavelet product integrals [NRH04, SM06], resulting in accurate all-frequency renderings of glossy materials and shadows under direct lighting, even for objects moving relative to each other.

Precomputed radiance transfer methods can be used for other purposes as well, such as interactive editing of materials. Sun *et al.* [SZC\*07] describe a technique that models light transport taking into account low-frequency indirect lighting with parameters of a linear BRDF space baked into the model, which allows users to navigate through this BRDF space during rendering. Ben-Artzi *et al.* [BAERD08] demonstrate interactive editing of materials with global illumination for fixed lighting conditions and viewpoints.

The first work to address mesh resolution in the context of precomputed radiance transfer is described by Křivánek *et al.* [KPŽ04]. They present a technique that uses an error metric defined over the surface of the scene, to refine the geometry in regions with high approximation error. Their method however is limited to diffuse materials. In recent work [LZT\*08], a method is described that samples the transfer function independently of mesh geometry. They are able to determine sampling resolution adaptively in a hierarchical fashion. It is unclear however how to extend their technique to an all-frequency technique such as described in Green *et al.* [GKMD06], which uses an extensive set of nonlinear parameters.

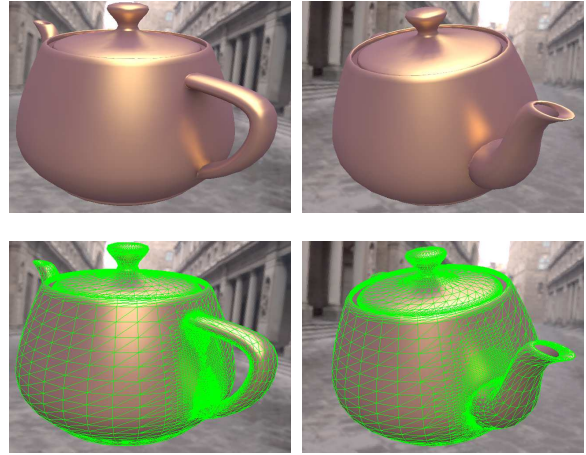
Our technique is most similar to the method proposed by Green *et al.* [GKMD06]. They model global light transport in a nonlinear fashion with a fixed set of spherical Gaussians, which can be efficiently interpolated to generate accurate glossy reflection of distant lighting. The parameters of these Gaussians are computed with a computationally expensive optimization procedure [CL96]. In our work we adjust this set of spherical lobes and the initial mesh adaptively. This allows us to render glossy interreflections accurately. Instead of spherical Gaussians we use the similar Von Mises-Fisher distribution [MJ00] because its parameters can be estimated with an efficient expectation-maximization procedure instead of a complex optimization procedure. The Von Mises-Fisher distribution has been used previously in computer graphics to model normal distributions [HSRG07].

### 3. Light Transport Model

The physical principles governing light transport are described by the rendering equation [Kaj86]. For non-emitting surfaces, the radiance  $L_o(\mathbf{x}, \mathbf{s}_o)$  reflected from surface point  $\mathbf{x}$  into direction  $\mathbf{s}_o$  is computed as follows:

$$L_o(\mathbf{x}, \mathbf{s}_o) = \int_{\Omega} f_r(\mathbf{s}_i, \mathbf{s}_o) L_i(\mathbf{x}, \mathbf{s}_i) \max(0, \mathbf{n}_x \cdot \mathbf{s}_i) d\omega_{\mathbf{s}_i}. \quad (1)$$

Here  $f_r(\mathbf{s}_i, \mathbf{s}_o)$  represents the BRDF and  $L_i(\mathbf{x}, \mathbf{s}_i)$  the light incident on  $\mathbf{x}$  from direction  $\mathbf{s}_i$ . Since all light originates from



**Figure 1:** The top row shows glossy interreflections on the body of the Utah Teapot from its handle and spout. In order to visualize such interreflections accurately, we precompute the light transport model adaptively by refining the initial mesh. The bottom row shows the final mesh of the Teapot.

the distant environment  $L_s(\mathbf{s}_i)$  and reaches  $\mathbf{x}$  directly or after one or more reflections, we define the transfer function  $T(\mathbf{x}, \mathbf{s}_i, \mathbf{s}_o)$  as describing how reflected light  $L_o(\mathbf{x}, \mathbf{s}_o)$  is composed of contributions from  $L_s(\mathbf{s}_i)$ :

$$L_o(\mathbf{x}, \mathbf{s}_o) = \int_{\Omega} T(\mathbf{x}, \mathbf{s}_i, \mathbf{s}_o) L_s(\mathbf{s}_i) d\omega_{\mathbf{s}_i}. \quad (2)$$

Precomputed radiance transfer techniques [SKS02] explicitly model this transfer function, which not only holds information about local reflection aspects such as material properties, but also about global lighting effects, such as indirect lighting and even subsurface scattering [WTL05].

Similar to the approach described by Green *et al.* [GKMD06], the view-dependent and the view-independent parts of the transfer function are treated separately:

$$T(\mathbf{x}, \mathbf{s}_i, \mathbf{s}_o) = T_d(\mathbf{x}, \mathbf{s}_i) + T_g(\mathbf{x}, \mathbf{s}_i, \mathbf{s}_o). \quad (3)$$

The view-dependent  $T_g(\mathbf{x}, \mathbf{s}_i, \mathbf{s}_o)$  is discretized with a finite set of surface points  $\mathbf{x} \in \mathbf{S}$  and directions  $\mathbf{s}_o \in \Omega$ . Each such two-dimensional transfer slice  $T_{g, \mathbf{x}, \mathbf{s}_o}(\mathbf{s}_i)$  is approximated with a sum of spherical lobes  $\gamma(\mathbf{s})$ :

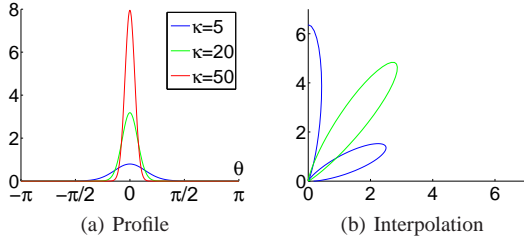
$$T_{g, \mathbf{x}, \mathbf{s}_o}(\mathbf{s}_i) \approx \sum_{j=1}^N \alpha_j \gamma_j(\mathbf{s}_i). \quad (4)$$

To model and render the view-independent part  $T_d(\mathbf{x}, \mathbf{s}_i)$ , any diffuse precomputed radiance transfer method can be used [SKS02, NRH03]. For simplicity, we treat this compo-

nent in the same manner as any other transfer slice (Equation 4).

Green *et al.* [GKMD06] choose an a priori set of viewing directions and surface points  $\mathbf{S} \times \Omega$ . However, glossy interreflections require this set to be very dense in some regions. Instead our method determines this set adaptively, so that glossy interreflections can be modeled accurately without redundant computation caused by a uniform dense sampling of  $\mathbf{S} \times \Omega$ . Furthermore we choose  $\gamma(\mathbf{s})$  to be the von Mises-Fisher distribution [MJ00]:

$$\gamma_j(\mathbf{s}_i) = c(\kappa_j) e^{\kappa_j(\mu_j \cdot \mathbf{s}_i)}. \quad (5)$$



**Figure 2:** (a) Profile of the von Mises-Fisher distribution. (b) For viewing directions that were not modeled explicitly, the transfer slice (green) can be computed by interpolating the  $\mu$  and  $\kappa$  parameters of nearby slices (blue).

The von Mises-Fisher (vMF) distribution (Figure 2a) is a spherical probability distribution which is nonlinearly dependent on the parameters  $\mu_j$  and  $\kappa_j$ :  $\mu_j$  is the unit mean direction of the lobe, while the scalar value  $\kappa_j$  determines its width. The normalization factor  $c(\kappa_j)$  is chosen such that the spherical integration of the vMF lobe equals 1. Our decision to use the von Mises-Fisher distribution is mainly motivated by the fact that efficient methods exist to estimate its parameters, which allows us to speed up precomputation significantly (see Section 5). Similarly to Green *et al.* [GKMD06] transfer slices for viewing directions  $\mathbf{s}_o$  and surface points  $\mathbf{x}$  that are not modeled explicitly, are computed by interpolating their respective  $\mu_j$  and  $\kappa_j$  parameters and weights  $\alpha_j$  (Figure 2b):

$$\tilde{\mu}_j = \frac{T(\mu_j)}{\|T(\mu_j)\|}, \tilde{\kappa}_j = T(\kappa_j), \tilde{\alpha}_j = T(\alpha_j). \quad (6)$$

#### 4. Adaptive Subdivision

Our method adaptively refines the angular and spatial resolution of the transfer function based on a measure that quantifies the similarity of two vMF distributions. We describe this measure in Subsection 4.1. During each iteration, the transfer function has to be computed in a number of surface points. For each of these points, the appropriate number of viewing directions  $\mathbf{s}_o \in \Omega$  is determined independently (Subsection 4.2). After each iteration, a new face is chosen to be subdivided, thereby creating new sample points  $\mathbf{x} \in \mathbf{S}$ . We

show where and how these faces are chosen in order to visualize glossy interreflections faithfully (Subsection 4.3).

#### 4.1. Difference Between Two vMF Distributions

We choose the spherical  $L_2$  distance to describe the difference between two vMF distributions. Due to the fact that these distributions integrate to one over the sphere, an approximate expression for the  $L_2$  distance between two vMF lobes can be derived easily:

$$\begin{aligned} & \|\alpha_0 \gamma_0(\mathbf{s}) - \alpha_1 \gamma_1(\mathbf{s})\|_2^2 \\ & \approx \frac{\alpha_0^2 \kappa_0}{4\pi} + \frac{\alpha_1^2 \kappa_1}{4\pi} - \frac{\alpha_0 \alpha_1 \kappa_0 \kappa_1}{\pi \kappa_c} e^{-\kappa_c - (\kappa_0 + \kappa_1)}, \end{aligned} \quad (7)$$

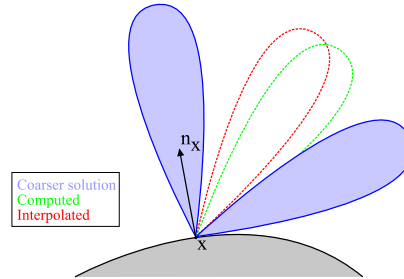
with

$$\kappa_c = \frac{1}{\|\mu_0 \kappa_0 + \mu_1 \kappa_1\|_2}. \quad (8)$$

This approximation is only valid for  $\kappa_0, \kappa_1 \gg 1$ , which always applies for our transfer functions ( $\min(\kappa) \approx 3$ ). A more detailed derivation of Equation 7 can be found in Appendix A.

#### 4.2. Angular Subdivision

For every surface point, we gradually add viewing directions determined through icosahedral subdivision and compute their corresponding transfer slices as described in Section 5. We continue this process as long as these slices improve the quality of the approximation significantly.



**Figure 3:** To determine whether additional transfer slices need to be computed, we compare new transfer slices with their predictions determined through interpolation.

For each icosahedral subdivision level  $\Omega_i$ , we not only compute the corresponding transfer slices, but also predict them by interpolating transfer slices for  $\Omega_0 \cup \dots \cup \Omega_{i-1}$  (Figure 3). We repeat this routine until the finest level of detail is reached or until the mean  $L_2$  distance between computed and predicted slices falls below a predetermined threshold value. In this manner we avoid aliasing artifacts in the presence of blocking geometry caused by undersampling  $\Omega$ , and we make sure no errors are introduced due to interpolating the nonlinear parameters  $\mu$  and  $\kappa$ .

### 4.3. Spatial Subdivision

Figure 4 shows the different steps of our spatial subdivision scheme. We initialize the precomputation by determining the transfer function at each vertex of the scene. Then our spatial difference metric can be applied. All triangles are sorted according to this metric and the triangle with the highest error is subdivided. We continue this process until there are no triangles left to subdivide.

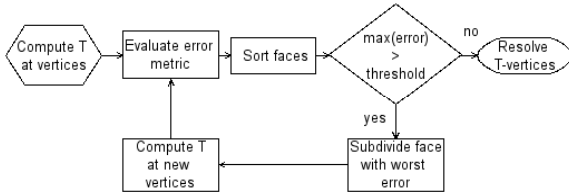


Figure 4: Flow chart of our spatial subdivision process.

We use a very simple midpoint mesh subdivision scheme which splits each triangle in 4 new triangles. We require that no two neighboring triangles can differ more than one subdivision level. This scheme thus allows only one T-vertex at each edge of a triangle. After the precomputation, all T-vertices are resolved as a postprocess.

Unlike in [GKMD06], we define viewing directions relative to the local frame aligned with  $\mathbf{n}_x$ . We then compute the difference between the transfer function at points  $\mathbf{x}$  and  $\mathbf{y}$  by computing the mean  $L_2$  distance of corresponding transfer slices. When the transfer function has a different angular resolution at  $\mathbf{x}$  and  $\mathbf{y}$ , missing transfer slices are determined through interpolation. The choice to define viewing directions in a local frame is motivated by the fact that the BRDF has the same shape at  $\mathbf{x}$  and  $\mathbf{y}$  for every  $\mathbf{s}_o$  in this manner. Thus the difference between the transfer function in two points is primarily dependent on the presence of blocking geometry, and less on surface orientation.

For each triangle we determine the difference between the transfer function computed at each pair of vertices. We subdivide a triangle until the maximum of these differences falls below a predetermined threshold. Only triangles with an area larger than a predefined minimum are considered for subdivision. At visualization time, for every point  $\mathbf{x}$  on face  $\mathbf{F} \subset \mathbf{S}$ , the transfer function must be determined through barycentric interpolation of the transfer function at the vertices of  $\mathbf{F}$ . With our spatial subdivision method we are able to prevent aliasing artifacts where the transfer function varies rapidly.

### 5. Computing a Transfer Slice

For each surface point  $\mathbf{x}$  and viewing direction  $\mathbf{s}_o$  we approximate the transfer function with a weighed sum of vMF distributions (Equation 4). We estimate their parameters by generating sampling distributions with a probability density function (PDF) proportional to each transfer slice  $T_{g,\mathbf{x},\mathbf{s}_o}(\mathbf{s}_i)$  (Subsection 5.1). We then process these samples and derive the parameters for the vMF distributions that can explain them most likely (Subsection 5.2).

### 5.1. Generating the Transfer Distribution

We need to generate a sampling distribution with a PDF proportional to each transfer slice  $T_{g,\mathbf{x},\mathbf{s}_o}(\mathbf{s}_i)$ . For most cases of practical interest however it is impossible to derive an analytical identity for this PDF. We simplify this problem by considering each kind of light path contribution separately. In this context, a light path is defined as a succession of an arbitrary number of diffuse and glossy reflections. Additionally –assuming isotropic BRDFs– we can approximate each cosine-weighted BRDF present in the scene as a combination of a diffuse and a glossy term for a fixed set of viewing directions, which are both fit with 1 vMF lobe:

$$f_{r,\mathbf{s}_o}(\mathbf{s}_i) \cos(\mathbf{s}_i) \approx \beta_d \gamma_d(\mathbf{s}_i) + \beta_g \gamma_g(\mathbf{s}_i). \quad (9)$$

We generate samples according to the cosine-weighted BRDF approximately by using Equation 9 as PDF. Importance sampling with the vMF PDF can be accomplished with the following identities:

$$\theta = \frac{\arccos(\ln(e^\kappa - 2u \sinh(\kappa)))}{\kappa}, \quad (10)$$

$$\phi = 2\pi v, \quad (11)$$

with  $u, v$  uniformly distributed in  $[0..1[$ . The circle-symmetric distributions generated with these identities have to be rotated into a local frame aligned with  $\mu$  afterwards. This is illustrated in figure 5. For  $\kappa \gg 1$ , Equation 10 can be simplified further to:

$$\theta = \frac{\arccos(\ln(1-u) + \kappa)}{\kappa} \quad (12)$$

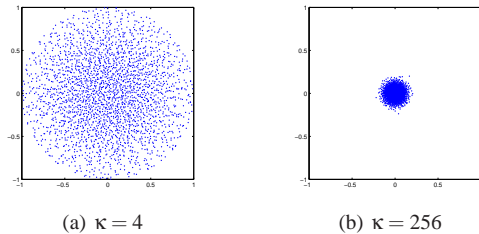
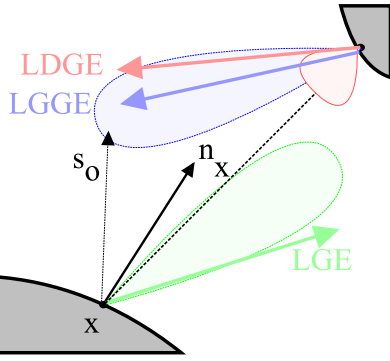


Figure 5: Sampling distributions, generated with Equations 10-11 for small and large values of  $\kappa$ .

For each  $\mathbf{x}$  and  $\mathbf{s}_o$ , we generate a certain number of samples according to the corresponding slice from Equation 9 with a basic path tracing technique (Figure 6). On intersecting the scene, an additional sample is recursively generated for both terms of Equation 9. Samples that are reflected into the distant lighting environment are stored for each kind of light path separately, to be used as input for the EM algorithm. Each set of samples thus describes the contribution to the reflected light on  $\mathbf{x}$  towards  $\mathbf{s}_o$  of the light following that specific path. For our experiments, we have only computed direct lighting and single-bounce interreflections,



**Figure 6:** We generate a sampling distribution for each light path contribution. We trace samples through the scene until they are reflected into the distant lighting environment. Then they are added to the appropriate distribution. We adapt Heckbert's notation [Hec90] to describe different light path contributions.

which leaves three light path contributions to be modeled for each viewing direction: direct, indirect diffuse and indirect glossy.

## 5.2. Estimating the Parameters

The Expectation-Maximization method is an iterative algorithm that determines a mixture of vMF distributions that explains an instantiated directional distribution. Each iteration consists of two steps. During the Expectation step, conditional probabilities are derived for each sample for each distribution in the von Mises-Fisher mixture, expressing the likelihood of being an instance of that distribution. Based on these probabilities, the parameters for each distribution are re-estimated during the Maximization step.

We choose to describe each kind of light path contribution with a single lobe. This assumption reduces the EM algorithm to an analytic description for the desired parameters instead of an iterative algorithm. Given  $M$  the total number of samples,  $\mathbf{n}_j$  the mean unnormalized direction of all  $M_j$  samples for light path contribution  $j$ ,  $l$  the number of reflections occurring on this path, and  $\beta_{i,j,k}$  the weight of the lobe sampled at the  $k$ th reflection of the  $i$ th sample, then:

$$\kappa_j = \frac{3\|\mathbf{n}\| - \|\mathbf{n}\|^3}{1 - \|\mathbf{n}\|^2}, \quad (13)$$

$$\mu_j = \frac{\mathbf{n}}{\|\mathbf{n}\|}, \quad (14)$$

$$\alpha_j = \frac{M_j}{M} \sum_{i=1}^{M_j} \left[ \prod_{k=1}^l \beta_{i,j,k} \right]. \quad (15)$$

Equations 13-14 are unbiased estimators for  $\mu$  and  $\kappa$  [BDGS05]. Equation 15 follows naturally from performing the spherical integration of each light path contribution given that paths are constructed by importance sampling either the glossy or the diffuse component of Equation 9.

Scene	e	Initial	Final	Time(hours)
teapot	4	2086	9727	2.1
teapot	16	2086	10496	3.6
teapot	32	2086	11646	4.1
teapot	64	2086	13724	5.3
teapot	128	2086	17146	21.8
buddha	64	13997	25998	10.3
Bunny	64	2602	10326	1.5

**Table 1:** The initial and final number of vertices and the total precomputation time (in hours) for each example. We generated five scenes including the Utah Teapot modeled with a Blinn-Phong BRDF (exponents 4, 16, 32, 64 and 128), a Stanford Buddha model (exponent 64) and a Stanford Bunny (exponent 64) on a diffuse plane.

We only use a single lobe per light path contribution for simplicity. In the case of highly glossy materials which have a rather narrow profile this approximation is usually sufficient. Of course this assumption is not valid for diffuse materials which give rise to more complex transfer functions. Although we have implemented the diffuse component of the transfer function with vMF distributions as well, any PRT technique could be used to render diffuse materials [SKS02, NRH03]. However it is possible to approximate each light path contribution with multiple lobes with the full iterative EM method. We leave this extension as future work.

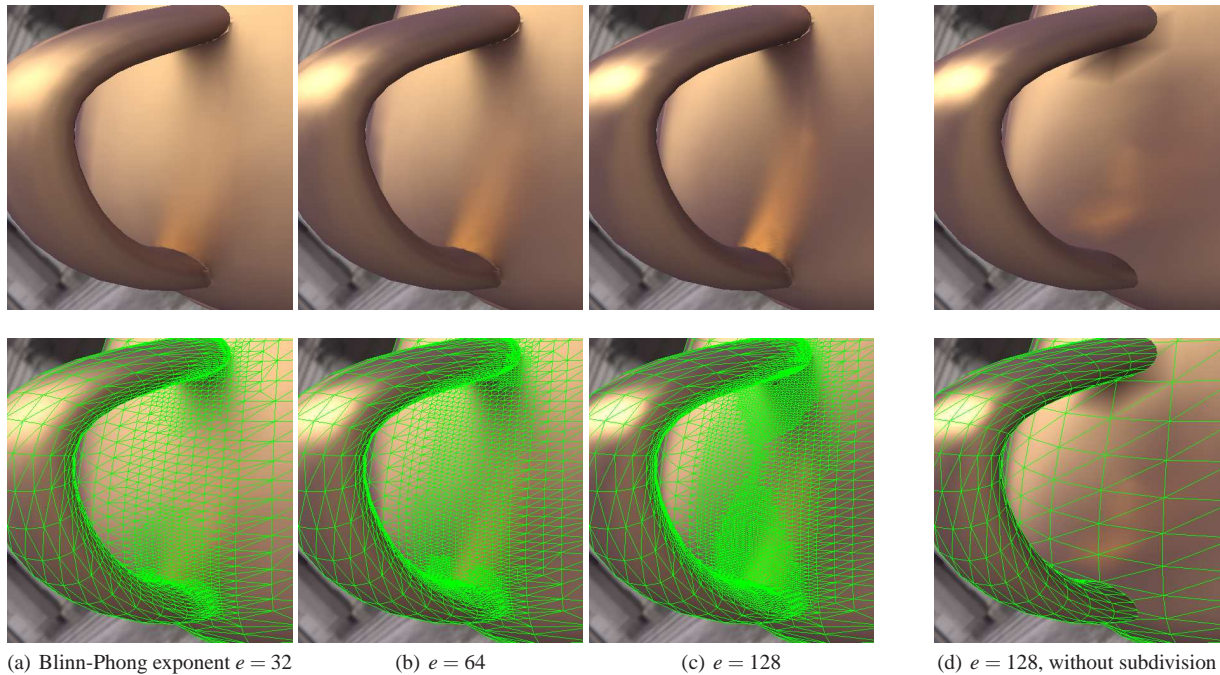
## 6. Results and Discussion

### 6.1. Precomputation

We start by computing the transfer function at each vertex of the initial mesh. Viewing directions and their corresponding transfer slices are added until the appropriate angular resolution is obtained. We allow a maximum of 5 subdivision levels. After this initialization, we sort faces according to our spatial error heuristic. All computations were done on a Dual Intel Xeon X5365 Quad Core 3GHz, with 16GB of memory. To utilize the parallel capacity of this system efficiently, we subdivide 200 faces before computing the transfer function during each iteration. Precomputation statistics for all our experiments can be found in Table 1.

To illustrate our refinement methods, we generated three identical teapot scenes with different variations of the Blinn-Phong reflectance model (Figure 7)a-c. The glossier the applied material, the more the mesh is subdivided by our method. In this manner, the reflection of the handle on the body of the teapot can be reproduced. Figure 7d shows the result for the teapot with Blinn-Phong exponent 128 for the initial mesh which allows us to compare our algorithm with [GKMD06]. In this case the transfer function is not sampled densely enough spatially for this interreflection to be visible. On the other hand a uniformly dense sampling would result in a model consisting of 163,256 vertices, whereas our adaptive approach requires only 17,146 vertices.

Figure 8 shows the angular resolution for every transfer function. The angular resolution is fairly constant for each material and increases with glossiness, although it is also dependent on the presence of occluders. However it is very



**Figure 7:** (a,b,c) Three Utah Teapot models with Blinn-Phong BRDF (exponents 32, 64 and 128). The amount of spatial subdivision increases with increasing glossiness, in order to represent the glossy reflections caused by the teapot’s handle accurately. (d) A Utah Teapot with Blinn-Phong BRDF (exponent 128) rendered with its initial tessellation, similar to [GKMD06]. In this case the transfer function is not sampled densely enough to represent the glossy interreflection of the handle.

important to represent the transfer function at the appropriate angular resolution because it allows us to compare the transfer function spatially at neighboring vertices. Under-sampling the transfer function angularly makes our spatial error heuristic unreliable. Both angularly and spatially we therefore use the same threshold value, which works well in practice for all our experiments.

Each material used in our precomputation has to be fit with vMF distributions for a number of viewing directions, determined through icosahedral subdivision (Equation 9). Slices for other viewing directions are obtained through barycentric interpolation of the 3 closest computed view-points. For a Blinn-Phong BRDF with glossy exponent  $e = 128$ , it takes about 47 seconds to compute all slice parameters with the nonlinear minimization procedure described in [CL96]. Fitting a similar transfer function with our EM estimation on the other hand, takes about 2 seconds. This means that without our estimation method it would take weeks to compute our results!

## 6.2. Rendering

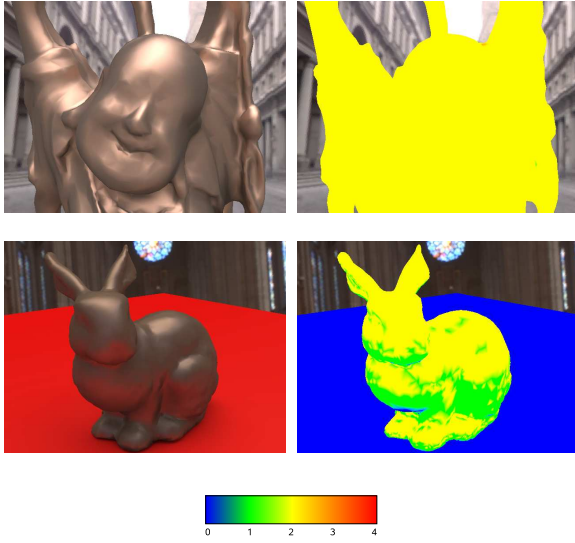
Rendering is done entirely on the GPU in vertex and pixel shaders. All scenes were rendered on a PC with an Intel Core 2 Duo X6800 2.93 GHz, 4GB of memory and an nVidia GeForce 8800GTX GPU. In a preprocess the environment map is prefiltered with several vMF profiles. The results are stored in a mipmap. Each level of the mipmap is filtered with

Scene	e	Size (MB)	Frame rate (FPS)
Teapot	4	25.3	453
Teapot	16	39.3	431
Teapot	32	45.6	403
Teapot	64	57.9	360
Teapot	128	256.2	281
Buddha	64	99.8	183
Bunny	64	19.0	300

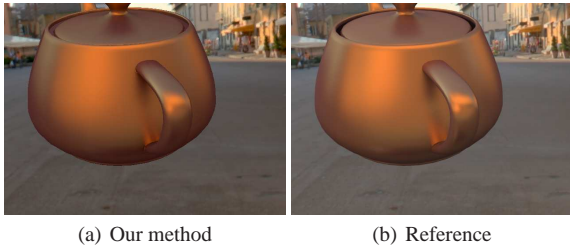
**Table 2:** Frame rate and size of precomputed data in texture memory (16-bit precision per floating point value) for each scene described in Table 2.

a vMF kernel such that the kernel’s  $\kappa$  value is doubled between successive levels [GKD07]. Then vertex shaders compute the appropriate transfer slices, which are interpolated during rasterization (Equation 6). Pixel shaders determine the final lighting contribution by indexing the mipmap. For each lighting contribution, the level of detail is determined by the contribution’s  $\kappa$  value.

Our implementation renders all our results at real-time frame rates (Table 2), and produces similar results to a ray-traced reference (Figure 9).



**Figure 8:** The number of angular subdivision levels for representing the transfer functions accurately. The top row shows the Buddha model: two angular subdivisions are needed for the corresponding transfer function. The bottom row shows a glossy bunny on a nearly diffuse plane: angular resolution increases with glossiness and is influenced by occlusions.



**Figure 9:** Our method compared with a raytraced reference.

## 7. Conclusion

In this paper, we have presented a technique for precomputing and visualizing glossy interreflections accurately. Two important contributions make this possible: our spatial and angular refinement schemes that minimize memory usage and precomputation times and our fast estimation of vMF distribution parameters of the different lighting contributions. We believe that this work provides valuable insights in the resolutions necessary to precompute glossy interreflections.

An important limitation of our method is its lack of data compression scheme. This problem could be solved by using our difference heuristic to estimate existing redundancy. We leave this for future work.

submitted to *COMPUTER GRAPHICS Forum* (6/2009).

## Acknowledgements

Jurgen Laurijssen is supported by grant CREA/08/017 from the Research Fund K.U.Leuven (Onderzoeksfonds K.U.Leuven).

## Appendix A: L2 distance between two VMF distributions

$$L_{VMF} = \| \alpha_0 \gamma_0(\mathbf{s}) - \alpha_1 \gamma_1(\mathbf{s}) \|_2^2 =$$

$$\alpha_0^2 c(\kappa_0)^2 \int_{\Omega} e^{2\kappa_0 \mu_0 \cdot \mathbf{s}} d\mathbf{s}$$

$$+ \alpha_1^2 c(\kappa_1)^2 \int_{\Omega} e^{2\kappa_1 \mu_1 \cdot \mathbf{s}} d\mathbf{s}$$

$$- 2\alpha_0 \alpha_1 c(\kappa_0) c(\kappa_1) \int_{\Omega} e^{(\kappa_0 \mu_0 + \kappa_1 \mu_1) \cdot \mathbf{s}} d\mathbf{s}$$

where  $c(\kappa)$  is the normalization factor for the VMF shape:

$$c(\kappa) = \frac{\kappa}{4\pi \sinh(\kappa)} \quad (16)$$

Thus the unnormalized shape integrates to  $\frac{1}{c(\kappa)}$ . By normalizing the  $\kappa_0 \mu_0 + \kappa_1 \mu_1$  vector, a valid  $\mu$  and  $\kappa$  is derived for the cross term (Equation 8):

$$L_{VMF} = \frac{\alpha_0^2 c(\kappa_0)^2}{c(2\kappa_0)} + \frac{\alpha_1^2 c(\kappa_1)^2}{c(2\kappa_1)} - \frac{2\alpha_0 \alpha_1 c(\kappa_0) c(\kappa_1)}{c(\kappa_c)}$$

Simple algebraic manipulation, taking into account that  $\sinh(2\kappa)$  equals  $2 \cosh(\kappa) \sinh(\kappa)$ , gives us the following result:

$$= \alpha_0^2 c(\kappa_0) \cosh(\kappa_0) + \alpha_1^2 c(\kappa_1) \cosh(\kappa_1) - \frac{2\alpha_0 \alpha_1 c(\kappa_0) c(\kappa_1)}{c(\kappa_c)}$$

For  $\kappa_0, \kappa_1 \gg 1$ , which always applies for our transfer functions ( $\min(\kappa) \approx 3$ ), our analytic formula can be further simplified:

$$= \frac{\alpha_0^2 \kappa_0}{4\pi} + \frac{\alpha_1^2 \kappa_1}{4\pi} - 2 \frac{\alpha_0 \alpha_1 \kappa_0 \kappa_1 e^{\kappa_c - (\kappa_0 + \kappa_1)}}{2\pi \kappa_c}$$

## References

- [BAERD08] BEN-ARTZI A., EGAN K., RAMAMOORTHI R., DURAND F.: A precomputed polynomial representation for interactive brdf editing with global illumination. *ACM Trans. Graph.* 27, 2 (2008), 1–13.
- [BDGS05] BANERJEE A., DHILLON I. S., GHOSH J., SRA S.: Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6 (2005), 1345–1382.
- [CL96] COLEMAN T. F., LI Y.: An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* 6 2 (1996), 418–445.

- [DBB06] DUTRÉ P., BALA K., BEKAERT P.: *Advanced Global Illumination, second edition*. A. K. Peters, Ltd., Natick, MA, USA, 2006.
- [DeV98] DEVORE R. A.: Nonlinear approximation. *Acta Numerica* 7 (1998), 51–150.
- [GKD07] GREEN P., KAUTZ J., DURAND F.: Efficient reflectance and visibility approximations for environment map rendering. *Computer Graphics Forum* 26, 3 (2007), 495–502.
- [GKMD06] GREEN P., KAUTZ J., MATUSIK W., DURAND F.: View-dependent precomputed light transport using nonlinear gaussian function approximations. In *13D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), ACM Press, pp. 7–14.
- [Hec90] HECKBERT P. S.: Adaptive radiosity textures for bidirectional ray tracing. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1990), ACM, pp. 145–154.
- [HSRG07] HAN C., SUN B., RAMAMOORTHY R., GRINSPUN E.: Frequency domain normal map filtering. *ACM Trans. Graph.* 26, 3 (2007), 28.
- [Kaj86] KAJIYA J. T.: The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1986), ACM Press, pp. 143–150.
- [KM99] KAUTZ J., MCCOOL M. D.: Interactive rendering with arbitrary brdfs using separable approximations. In *Proceedings of the 10th Eurographics Workshop on Rendering* (1999), pp. 281–292.
- [KPŽ04] KRIVÁNEK J., PATTANAIK S., ŽÁRA J.: Adaptive mesh subdivision for precomputed radiance transfer. In *SCCG '04: Proceedings of the 20th spring conference on Computer graphics* (New York, NY, USA, 2004), ACM Press, pp. 106–111.
- [KSS02] KAUTZ J., SNYDER J., SLOAN P.-P.: Fast arbitrary brdf shading for low-frequency lighting using spherical harmonics. In *13th Eurographics Workshop on Rendering* (Pisa, Italy, 2002), Debevec P., Gibson S., (Eds.), Eurographics Association.
- [LK03] LEHTINEN J., KAUTZ J.: Matrix radiance transfer. In *13D '03: Proceedings of the 2003 symposium on Interactive 3D graphics* (New York, NY, USA, 2003), ACM Press, pp. 59–64.
- [LSS04] LIU X., SLOAN P.-P., SHUM H.-Y., SNYDER J.: All-Frequency Precomputed Radiance Transfer for Glossy Objects. In *Eurographics Symposium on Rendering* (Norrköping, Sweden, 2004), Keller A., Jensen H. W., (Eds.), Eurographics Association, pp. 337–344.
- [LZT\*08] LEHTINEN J., ZWICKER M., TURQUIN E., KONTKANEN J., DURAND F., SILLION F., AILA T.: A meshless hierarchical representation for light transport. In *ACM Transactions on Graphics* 27(3) (*Proc. SIGGRAPH 2008*), to appear (New York, NY, USA, 2008), ACM Press.
- [MJ00] MARDIA K. V., JUPP P. E.: *Directional Statistics*. John Wiley and Sons Ltd., 2000.
- [MTR08] MAHAJAN D., TSENG Y.-T., RAMAMOORTHY R.: An Analysis of the In-Out BRDF Factorization for View-Dependent Relightings. In *Proceedings of the 10th Eurographics Workshop on Rendering* (2008), pp. 1137–1145.
- [NRH03] NG R., RAMAMOORTHY R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 376–381.
- [NRH04] NG R., RAMAMOORTHY R., HANRAHAN P.: Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph.* 23, 3 (2004), 477–487.
- [SHHS03] SLOAN P.-P., HALL J., HART J., SNYDER J.: Clustered principal components for precomputed radiance transfer. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM Press, pp. 382–391.
- [SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2002), ACM Press, pp. 527–536.
- [SM06] SUN W., MUKHERJEE A.: Generalized wavelet product integral for rendering dynamic glossy objects. *ACM Trans. Graph.* 25, 3 (2006), 955–966.
- [SZC\*07] SUN X., ZHOU K., CHEN Y., LIN S., SHI J., GUO B.: Interactive relighting with dynamic brdfs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers* (New York, NY, USA, 2007), ACM, p. 27.
- [TS06] TSAI Y.-T., SHIH Z.-C.: All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers* (New York, NY, USA, 2006), ACM Press, pp. 967–976.
- [WTL04] WANG R., TRAN J., LUEBKE D.: All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In *Eurographics Symposium on Rendering* (Norrköping, Sweden, 2004), Keller A., Jensen H. W., (Eds.), Eurographics Association, pp. 345–354.
- [WTL05] WANG R., TRAN J., LUEBKE D.: All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph.* 24, 3 (2005), 1202–1207.
- [WTL06] WANG R., TRAN J., LUEBKE D.: All-frequency relighting of glossy objects. *ACM Trans. Graph.* 25, 2 (2006), 293–318.