*To appear in the proceedings of the 2006 Symposium in Interactive 3D Graphics*

# View-Dependent Precomputed Light Transport Using Nonlinear Gaussian Function Approximations

Paul Green[1] Jan Kautz[1] Wojciech Matusik[2] Frédo Durand[1]

[1]Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

[2]Mitsubishi Electric Research Laboratory
(MERL)

## Abstract

We propose a real-time method for rendering rigid objects with complex view-dependent effects under distant all-frequency lighting. Existing precomputed light transport approaches can render rich global illumination effects, but high-frequency view-dependent effects such as sharp highlights remain a challenge. We introduce a new representation of the light transport operator based on sums of Gaussians. The nonlinear parameters of our representation enable 1) arbitrary bandwidth because scale is encoded as a direct parameter, and 2) high-quality interpolation across view and mesh triangles because we interpolate the mean direction of the Gaussians, thereby preventing linear cross-fading artifacts. However, fitting the precomputed light transport data to this new representation requires solving a nonlinear regression problem that is more involved than traditional linear and nonlinear (truncation) approximation techniques. We present a new data fitting method based on optimization that includes energy terms aimed at enforcing artifact-free interpolation. We demonstrate that our method achieves high visual quality with a small storage cost and an efficient rendering algorithm.

**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:** Precomputed Radiance Transfer, Reflectance Field, Real-Time Rendering

Figure 1: A bird model with spatially varying material properties. The material transforms from a diffuse yellow material at the feet to a highly specular green material at the head. The tessellation of the model is low (∼12K vertices), nevertheless the reflections contain high frequencies due to our novel interpolation scheme.

## 1 Introduction

The quality of real-time rendering can be dramatically improved by precomputing the interaction of an object with light. This however raises a formidable challenge, the storage and re-use of data over a six-dimensional domain: two dimensions for each of lighting (assumed to be at infinity), view direction, and surface position. Current methods address this issue with compromises between bandwidth (or sharpness) of the illumination effects, storage costs, and rendering time. In this paper we focus on spatially varying view-dependent effects, such as glossy reflections, since view-independent effects like shadows can be handled with known techniques [Ng et al. 2003; Sloan et al. 2003]. View-dependent effects are harder to address because they involve variation over the full 6D domain, and highlights can vary quickly over space and view direction, which may result in cross-fading artifacts when linear interpolation is used between views or mesh vertices. We introduce a method that affords arbitrary bandwidth for view-dependent effects with a compact storage and real-time performance, at the cost of a more involved preprocess.

Precomputed light transport techniques compute a pixel color at render-time based on a precomputed approximation of the light transport operator, which evaluates the contribution of all lighting directions for a given spatial location on a mesh and a given view direction. Typically, the transport operator is represented by linear-basis functions such as spherical harmonics or wavelets, and the number of coefficients directly constrains the sharpness of the effects that can be handled. Compression techniques such as the nonlinear truncation of linear basis [Ng et al. 2003] or the use of separable approximations [Liu et al. 2004; Wang et al. 2004] improve on both storage and computation, but the number of coefficients required to handle high-frequency effects remains large and the rendering cost is directly proportional.

In our work, we address these issues and present a technique that can achieve high-quality view-dependent lighting effects with a small storage cost. We do not consider the view-independent component of light transport because it can be handled by well-known techniques [Ng et al. 2003; Sloan et al. 2003]. Our contribution lies in specular effects, where we represent the view-

dependent component of the transport operator using sums of Gaussian functions defined on the sphere of directions corresponding to the environment lighting. The spherical Gaussians are parameterized by a RGB color, mean direction, and variance. The nonlinear effect of the variance parameter allows for arbitrary scaling of the kernels, thereby affording all-frequency effects. The nonlinear effect of the mean direction parameter permits smooth interpolation across space (mesh triangles) and view direction, and prevents cross-fading artifacts, in the spirit of Phong interpolation. We typically use a small number of Gaussians, between one and three. Finally, the representation can be easily evaluated at runtime using a mip-mapped prefiltered environment map where each level corresponds to a Gaussian variance. However, the price to pay is a more involved preprocess, since nonlinear function fitting is not as straightforward as linear projection or basis truncation. To address this challenge, we introduce an optimization technique that achieves a good approximation as well as high-quality interpolation over view directions and mesh triangles. In the end, the per-pixel interpolation of the Gaussian direction and variance is critical to achieve high visual quality.

The three major contributions of this work are:

- We introduce a nonlinear representation for light transport based on a sum of Gaussians that is compact and leads to a fast integration with the lighting at render time.
- Our representation affords high-quality interpolation across mesh location and view direction, which facilitates rendering from arbitrary viewpoints and requires only sparse sampling.
- We present an optimization approach to fitting Gaussian function approximations to precomputed light transport data. In particular, we include energy terms to favor artifact-free interpolation.

# 2 Background

## 2.1 Related Work

Our work draws from a number of areas in computer graphics: scene relighting, environment mapping and prefiltering methods, radiance transfer methods, and acquisition techniques. Our work is also related to the nonlinear Lafortune BRDF model [1997], and it requires a similar nonlinear regression.

**Scene Relighting** allows the user to relight a fixed scene. Conceptually, scene relighting algorithms precompute a separate global illumination solution per light source. Linear combination of these results provides limited dynamic effects. Early work [Dorsey et al. 1991] adjusts intensities of a fixed set of sources and is not intended to fit general lighting environments. Nimeroff et al. [1994] and Teo et al. [1997] generalize to more flexible emitters but are still limited to a fixed view. Dobashi et al. [1995] use the spherical harmonics basis and transfer vector fields over surfaces to allow viewpoint change but restrict lighting changes to the directional intensity distribution of an existing set of non-area light sources in diffuse scenes. Debevec et al. [2000] relight faces using a directional light basis.

The concept of relighting has also been used for texture maps. Malzbender et al. [2001] precompute a higher-dimensional texture, called a *polynomial texture map*, which allows real-time interreflection effects as well as shadowing. A similar approach using a steerable basis for directional lighting is used in [Ashikhmin and Shirley 2002]. None of these methods can deal with high-frequency reflections and offer only restricted view-dependent effects.

**Environment Map Prefiltering** techniques proposed by Greene [1986] observed that an environment map prefiltered by a BRDF could be used to simulate diffuse and glossy reflections. The original environment mapping work [Blinn and Newell 1976] can also be cast as prefiltering, by considering specular reflection as a convolution of the environment map with an impulse function. There has been a large amount of related research using prefiltered environment maps [Kautz 2004; McAllister et al. 2002], however none of these methods handle shadowing or indirect lighting. McAllister et al. [2002] store spatially varying parameters of a Lafortune BRDF model in texture maps and use prefiltered environment maps to evaluate per-texel shading. Due to limitations of contemporary hardware (e.g., NVIDIA GeForce 4) they were unable to demonstrate a hardware implementation of rendering with environment lighting.

**Reflectance Prefiltering** was pioneered by Fournier [1992]. He uses nonlinear optimization to approximate distributions of microfacet normals at multiple resolutions using a sum of cosine lobes. Recently, Tan et al. [2005] proposed a method for multiresolution reflectance filtering using a Gaussian Mixture Model. They use the well known EM algorithm [Dempster et al. 1977] to estimate model parameters. While their target application is different, their work has several similarities to our method. One similarity is the need for correspondences between interpolated Gaussians. However, they modify the EM algorithm to include priors which favor correspondences, while we have chosen to use an optimization approach because it provides us flexibility to directly incorporate various coherence terms. The use of EM in our context is an interesting avenue of future work.

**Precomputed Radiance Transport** methods permit the relighting of an object using full spherical incident lighting. Sloan et al. [2002] use the spherical harmonics basis to represent how an object casts shadows and interreflections onto itself, called *precomputed radiance transfer*. This work is limited to diffuse reflectors or to surfaces with Phong-like glossy reflections. Improvements have been presented to incorporate more complex BRDFs [Kautz et al. 2002; Lehtinen and Kautz 2003], but the use of the spherical harmonics basis limits these approaches to low-frequency incident lighting. Even with compression [Sloan et al. 2003] the lighting cannot be high-frequency.

Ng et al. [2003] use nonlinear approximation (truncation) of wavelets as basis functions instead of spherical harmonics. With as few as 100 coefficients, they are able to incorporate high-frequency lighting effects (for diffuse surfaces or static views only). The significant coefficients are selected based on the incident lighting, which prevents the use of highly glossy surfaces. Arbitrary BRDFs can be incorporated using a method to evaluate triple product integrals [Ng et al. 2004], but is limited to direct lighting only. Arbitrary BRDFs and wavelets can be combined using separable BRDF approximations [Liu et al. 2004; Wang et al. 2004]. These techniques are limited to a small number of fixed BRDFs because precomputation is done on a per-BRDF basis (multiple BRDFS also incur additional memory consumption and storage costs). Furthermore, high-frequency BRDFs can only be represented using prohibitively many terms in the separable approximation [Liu et al. 2004].

The closest PRT technique to our work was proposed by Sloan et al. for local deformations [2005]. They represent the local effect of light transport using a weighted sum of rotated zonal harmonics. This nonlinear approximation to the transport is fitted using a greedy method combined with a local BFGS optimization step [Press et al. 1992]. Interestingly, the main goal of their work is the ability to efficiently rotate the representation in order to handle geometric deformation, albeit in the domain of low frequency lighting and transport. In contrast, we focus on the representation of arbitrary-frequency view-dependent effects. The arbitrary rotation of zonal harmonics is similar to the encoding of the mean direction of Gaussians in our method. Although this arbitrary rotation pro-

vides the possibility for similar interpolation gains, they do not take advantage of this in their work. Our approach also encodes scale as a nonlinear parameter, while zonal harmonics represent different frequencies using a linear sum of basis functions. To further emphasize their restriction to low frequencies, we point out that their lighting-transport integration is computed in the spherical harmonic basis, which has been shown [Ng et al. 2003] to require large numbers of coefficients to represent all-frequency lighting content. Fundamentally they are limited to the same class of low frequency transport functions as spherical harmonics.

The visual quality of all of the above techniques depend heavily on the tessellation of the model; high frequency effects require a fine tessellation or cross-fading artifacts are visible. Our method prevents this through nonlinear interpolation of transport functions instead of using linear blending.

**Acquisition Methods** deal with similar data, but instead of precomputing transport information, they infer it from real scenes. These approaches are related to our work because a number of them use nonlinear function approximation. We focus on such methods and omit techniques that deal with the inverse problem. The initial environment matting technique [Zongker et al. 1999] describes the observation in each pixel using a single 2D box function. This leads to a compact nonlinear representation based on the box coordinates and weight. This algorithm works well for specular and refractive objects. Higher quality is achieved by replacing the box functions with oriented 2D Gaussian kernels [Chuang et al. 2000].

The above methods assume a fixed viewpoint, i.e., their light transport representations are tied to a 2D plane. A hybrid method [Matusik et al. 2002] combines a forward method [Debevec et al. 2000] for low-frequency components and high-quality environment matting [Chuang et al. 2000] for high frequency reflections and refractions. Data is sampled from multiple viewpoints and combined with 3D geometry. Only one Gaussian is used for each type of allowed transport (i.e., reflection and refraction), which is interpolated for in-between views. Rendering with this representation is done in software and does not achieve interactive rates.

### 2.2 Review of Light Transport

Under the assumption that our object is not emissive, it is common to view light transport as a linear integral operator [Sloan et al. 2002; Ng et al. 2003]. The light that arrives at our eye is a linear transformation of the distant incident lighting. This operator $\mathscr{T}$, which we call the light transport operator, encodes the effects of the material properties and light transport interactions between different patches of the object. Thus we can describe the exit radiance $L_o(\mathbf{x},\omega_o)$ at a point $\mathbf{x}$ (on the object) along the outgoing (viewing) direction $\omega_o$ as the result of applying our integral operator $\mathscr{T}$ on the (distant) incident lighting $L_i$:

$$L_o(\mathbf{x},\omega_o) = (\mathscr{T}L_i)(\mathbf{x},\omega_o) = \int_S T_{\mathbf{x},\omega_o}(\omega_i) L_i(\omega_i) \mathrm{d}\omega_i, \qquad (1)$$

where $T_{\mathbf{x},\omega_o}$ is the integration kernel of $\mathscr{T}$, also sometimes called the *transport function*. For fixed $\mathbf{x}$ and $\omega_o$, $T_{\mathbf{x},\omega_o}$ is a 2D function parameterized over the sphere $S$ of incoming directions. It describes, for each direction $\omega_i$, the contribution of $L_i(\omega_i)$ to the total reflected radiance leaving $\mathbf{x}$ along $\omega_o$. Substituting for the integration kernel we can derive the familiar reflectance equation for direct lighting. With

$$T_{\mathbf{x},\omega_o}(\omega_i) = f_r(\omega_i,\omega_o) V_{\mathbf{x}}(\omega_i)(\mathbf{n}_{\mathbf{x}} \cdot \omega_i), \qquad (2)$$

where $\omega_i$ is the incoming direction, $\mathbf{n}_{\mathbf{x}}$ is the surface normal at $\mathbf{x}$, $V_{\mathbf{x}}$ is a binary visibility function, $f_r$ is the bidirectional reflectance distribution function (BRDF), we get:

$$L_o(\mathbf{x},\omega_o) = \int_\Omega f_r(\omega_i,\omega_o) V_{\mathbf{x}}(\omega_i)(\mathbf{n}_{\mathbf{x}} \cdot \omega_i)\, L_i(\omega_i) \mathrm{d}\omega_i\,. \qquad (3)$$

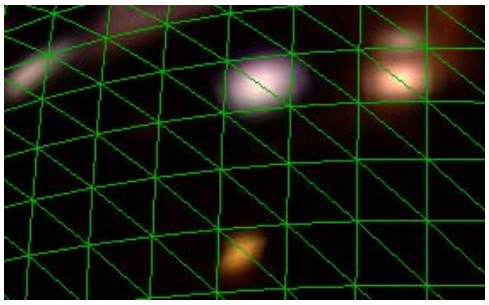


Figure 2: This close-up of a sphere shows that the highlights contain detail finer than the tessellation of the sphere would allow without our per-pixel Gaussian interpolation.

However, the integration kernel $T_{\mathbf{x},\omega_o}$ is not limited to direct illumination, and can describe many other complex transport effects such as interreflections, refraction, self-occlusion, caustics, subsurface scattering, and other indirect lighting.

Unfortunately, direct evaluation of $\mathscr{T}L_i$ (Equation 1) is infeasible (even assuming distant lighting, light transport is 6D) and fundamentally all previous PRT work is concerned with approximating and compressing some slice of the light transport operator. The rest of this paper describes new methods for computing and approximating the integration kernels $T_{\mathbf{x},\omega_o}$, as well as efficiently integrating Equation 1.

## 3 Method

As previously noted, the key issue common to all PRT techniques is the representation of the light transport operator. There are several criteria used to judge the quality of any proposed representation: rendered visual quality, compactness, efficient integration/evaluation, angular and spatial frequency bandwidth. Our new representation is based on Gaussian function approximations. We first introduce the representation for a given sample on the mesh and a given view direction. We next show how this representation affords high-quality interpolation between views and spatial locations. We discuss the challenges raised by the estimation of the model parameters and outline our optimization approach. Finally, we show how evaluation at render time reduces to a texture lookup in a prefiltered environment map.

### 3.1 Light Transport Representation

We use a nonlinear model to represent the light transport integration kernel $T_{\mathbf{x},\omega_o}$. In contrast to linear approximation, where all data is approximated by a projection onto a fixed set of basis functions (e.g., fifth-order spherical harmonics), the parameters of our model have a nonlinear effect on the approximation. This is a significant departure from the previous work based on nonlinear approximation [Ng et al. 2003]; that starts with a linear basis (e.g., Haar wavelets), and truncates small coefficients to zero. We approximate $T_{\mathbf{x},\omega_o}$ as a a sum of $N$ weighted isotropic Gaussians:

$$\tilde{T}_{\mathbf{x},\omega_o}(\omega_i) = \sum_k^N w_k G(\omega_i;\mu_k,\sigma_k), \qquad (4)$$

where $G$ is a 2D spherical Gaussian centered around $\mu_k$ with standard deviation $\sigma_k$ and weight $w_k$ .

Our choice of a nonlinear model such as Gaussians is motivated by three main properties: arbitrary frequency, accurate approximation with only small number of coefficients, and high visual quality

of the interpolation between Gaussians. In return, they require nonlinear regression for the estimation of the parameters, as described in Section 4. Nonlinear models such as Gaussian functions have the potential to better fit data with a small number of coefficients when their functional form is well-adapted to the problem at hand. In our case, we have chosen Gaussians because they are smooth, have a compact set of parameters, have arbitrary bandwidth, and approximate rendering data such as BRDFs well. The nonlinear effect of the variance $\sigma_k$ parameter allows for direct encoding of scale, thereby affording arbitrary bandwidth. The nonlinear effect of the mean direction $\mu_k$ permits better interpolation and prevents cross-fading artifacts (see Figure 2). However, when using multiple Gaussians, good correspondences are needed for artifact-free interpolation. We discuss our solution below.

### 3.2 Sparse Representation & Interpolation

We have described how Gaussian functions can approximate the 2D light transport integration kernel $T_{\mathbf{x},\omega_o}$ at a given point $\mathbf{x}$ for a given view direction $\omega_o$. Conveniently, it also enables a sparse representation in the remaining four dimensions for space and view direction. Key to this is the fact that interpolation of the Gaussian parameters closely matches the behavior of integration kernels. In particular, the nonlinear parameters $\mu_k$ and $\sigma_k$ preserve the qualitative shape of the kernel across interpolation which leads to a good visual reproduction of highlights. The quality gained compared to the linear interpolation of the highlights is similar to the difference between Gouraud and Phong shading.

In practice, we compute and approximate the transport functions $\tilde{T}_{\mathbf{x},\mathbf{v}_j}$ only at mesh vertices $\mathbf{x}$ and only for a set of 92 fixed viewing directions $V = \{\mathbf{v}_j\}$, obtained through subdivision of an icosahedron. Transport functions for other views or positions are not represented explicitly; instead, we interpolate them from nearby samples.

### 3.3 Model Estimation

Our model requires nonlinear parameter estimation for the variance, weight, and mean of the Gaussians for each view direction and spatial location. We introduce an optimization technique that minimizes the $L^2$ error with the data, and also includes energy terms to improve correspondences between Gaussian parameters of neighboring views and mesh vertices. Our nonlinear parameter estimation is described in Section 4.

### 3.4 Method Overview

An overview of our method can be found in Figure 3. During the preprocessing step, we simulate light transport. We then fit our Gaussian model using a new optimization approach described in Section 4. We store Gaussian parameters for each vertex and view direction.

At runtime, rendering is straightforward. The approximated and interpolated transport function $\tilde{T}_{\mathbf{x},\mathbf{v}}$ is integrated against the lighting, using a small number of texture accesses. This is efficient because the integration between the lighting and Gaussians at various locations and scales can be precomputed and stored in a mip-mapped environment map.

## 4 Precomputation

Our precomputation proceeds in two steps: we first compute light-transport data for all views and mesh vertices, and we next fit our Gaussian parameters using an optimization approach. The first step

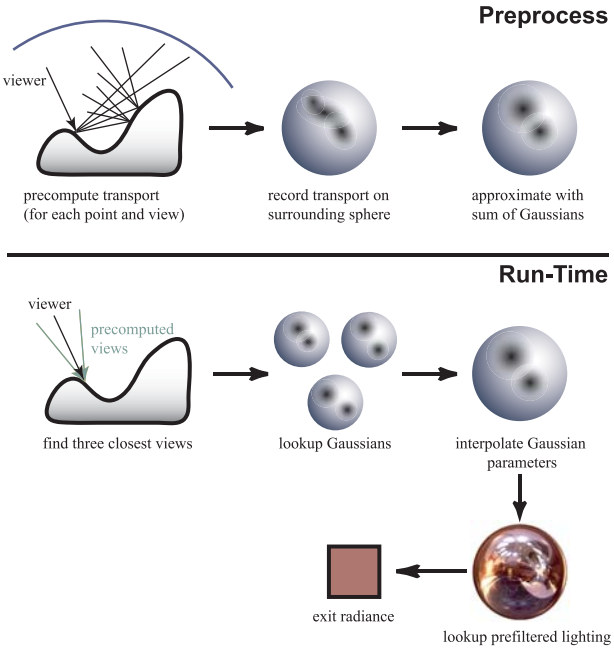



Figure 3: An overview of our method. We estimate the transport function for each view and vertex and approximate it with a sum of Gaussians. At runtime we interpolate the parameters from the three closest precomputed views to the current view and perform a simple lookup in a Gaussian-prefiltered environment map.

is not specific to our technique and any rendering approach could be used. Our contribution lies in the fitting of these data.

### 4.1 Light Transport Data

We compute $T_{\mathbf{x},\omega_o}$ for each view $\omega_o$ and for each vertex $\mathbf{x}$ of the model. In practice, we decompose $T_{\mathbf{x},\omega_o}$ as a sum of three terms, direct glossy, indirect glossy, and diffuse:

$$T_{\mathbf{x},\omega_o} = T^G_{\mathbf{x},\omega_o} + T^{IG}_{\mathbf{x},\omega_o} + T^D_{\mathbf{x},\omega_o}\,. \tag{5}$$

The diffuse component $T^D_{\mathbf{x},\omega_o}$ can include direct and indirect diffuse light transport. We do not discuss diffuse transport as it is a view-independent effect and we handle it using techniques that operate in the reduced 4D domain [Sloan et al. 2002; Ng et al. 2003]. For high-quality data, we use an appropriate precomputation method for each component (a common approach in path tracing implementations [Jensen 2001]). The direct glossy and diffuse components are evaluated for each texel of a cubemap in a fragment shader. An indirect glossy component is computed by BRDF importance sampled path tracing along each direction $\mathbf{v} \in V$ onto point $\mathbf{x}$. We record the path of each photon until it is absorbed or reflected off to infinity, where it is stored in the corresponding cubemap texel. Each of these processes produces a set of tabulated transport operators $T_{\mathbf{x},\omega_o}$, sampled over the texels of a cubemap. These cubemaps are the input to the next stage of precomputation, and the core of our technique: nonlinear Gaussian parameter estimation by optimization.

### 4.2 Approximating Light Transport

The main goal of our nonlinear optimization is to find the Gaussian parameters that best fit the light transport data. In addition, when using multiple Gaussians to encode the same component, we must

ensure good correspondences across views and mesh vertices. In particular, note that from a data fitting perspective, any permutation of $N$ Gaussians yields the same error. With this goal, we introduce additional energy terms that favor correspondence.

In practice, we concurrently fit the Gaussians across all views $\omega_{o,v}$ for a given vertex $\mathbf{x}$ using nonlinear optimization [Coleman and Li 1996] minimizing the following objective function:

$$\min_{\mu_k^{\omega_{o,v}}, \sigma_k^{\omega_{o,v}}, w_k^{\omega_{o,v}}} \sum_v \alpha D + \beta M + \gamma V + \delta S, \tag{6}$$

where $k = 1 \ldots N$ is one of the Gaussians, $v$ is one of the 92 views, and $\mu_k$, $\sigma_k$, and $w_k$ are the Gaussian parameters to be optimized. $D$ is a data-fitting term, and the following terms favor correspondences for the mean and variance across views, and finally across spatial locations, as we describe below.

**Data Term.** The data term is designed to minimize the $L^2$ error between the transport data and our Gaussian approximation:

$$D \equiv \sum^{S} \left\| \left[ \sum_{k=1}^{N} w_k G(\omega_i; \mu_k^{\omega_{o,v}}, \sigma_k^{\omega_{o,v}}) \right] - T_{\mathbf{x},\omega_{o,v}}(\omega_i) \right\|^2 . \tag{7}$$

**Alignment of Means.** Because we want to interpolate Gaussians across views, we add a mean alignment term to encourage corresponding Gaussians in neighboring views have consistent means:

$$M \equiv \sum_{k=1}^{N} \left\| \left[ \frac{1}{R} \sum_{r(v)} \mu_k^{\omega_{o,r(v)}} \right] - \mu_k^{\omega_{o,v}} \right\|^2 , \tag{8}$$

where $r(v)$ are the neighboring views of $v$. This term guides the mean for view $v$ toward the average mean of the neighboring views. This term is necessary to fill in data for occluded views (i.e., views with zero transport due to visibility, or views just below the horizon) in order to provide smooth interpolation at nearby views with nonzero transport functions.

**Alignment of Standard Deviation.** This term is similar to the one above; encouraging the standard deviation of the Gaussian in view $v$ to be similar to the average standard deviation in neighboring views:

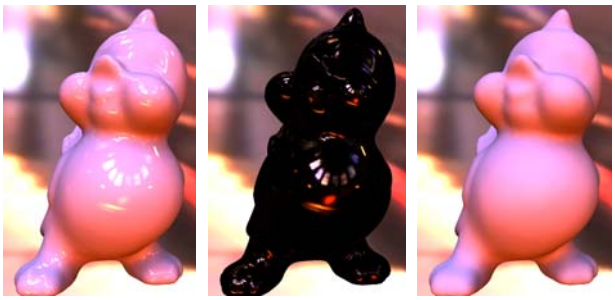
$$V \equiv \sum_{k=1}^{N} \left\| \left[ \frac{1}{R} \sum_{r(v)} \sigma_k^{\omega_{o,r(v)}} \right] - \sigma_k^{\omega_{o,v}} \right\|^2 . \tag{9}$$

**Spatial Term.** The spatial term ensures that the Gaussians of the current vertex (for all views) have good corresponding Gaussians at (already processed) neighboring vertices. This term aims to make the *relative* arrangement of spatial neighbors consistent:

$$S \equiv \sum_{k_1=1}^{N-1} \sum_{k_2=k_1}^{N} \sum_{s(\mathbf{x})} \max\left( (-\langle \mu_{\mathbf{x},k_1}^{\omega_{o,v}} - \mu_{\mathbf{x},k_2}^{\omega_{o,v}}, \mu_{s(\mathbf{x}),k_1}^{\omega_{o,v}} - \mu_{s(\mathbf{x}),k_2}^{\omega_{o,v}} \rangle)^3, 0 \right), \tag{10}$$

where $\mathbf{x}$ is the current vertex and $s(\mathbf{x})$ are the neighboring vertices that have already been processed. The relative arrangement is calculated by taking difference vectors between the Gaussian means $\mu$ for all pairs of the N Gaussians of a given view and vertex. The dot product between the difference vectors of neighboring vertices is positive when the relative arrangement is consistent, and negative when it is flipped.

Optimizing for the Gaussian parameters of all views and all vertices at the same time is impossible due to the very large number of variables. Instead, we only concurrently fit the parameters of a single vertex (all views). To make sure that spatial consistency is attained nonetheless, we proceed as follows. We start with a single vertex and optimize all its Gaussian parameters. After all its parameters have been optimized, we process a neighboring vertex using the above spatial consistency term. We continue processing neighboring vertices until all vertices have been processed (which essentially "flood-fills" the mesh).



(a) (b) (c)

Figure 4: Decomposition of a rendered frame (a) into the glossy component (b) and the diffuse component (c).

Tan et al. [2005] solve a similar problem for multiresolution reflectance data using the EM algorithm. EM is an iterative, nonlinear optimization routine effective for estimating parameters of a Gaussian Mixture Model. By incorporating prior constraints, they estimate model parameters as well as resolve correspondences between Gaussian parameters at neighboring texels. In contrast, we chose an optimization approach because of the flexibility to directly control the objective function. In the future, we hope to adapt the EM algorithm for our purpose.

## 5 Rendering

Rendering an object involves integrating the lighting $L_i(\omega_i)$ with the (interpolated) transport functions $\tilde{T}_{\mathbf{x},\mathbf{v}}(\omega_i)$, see Equation 1. This integration has to be repeated for every visible point $\mathbf{x}$:

$$L_o(\mathbf{x}, \omega_o) = \int_S \left( \sum_k^N w_k G(\omega_i; \mu_k, \sigma_k) \right) L_i(\omega_i) \mathrm{d}\omega_i . \tag{11}$$

We use well-known techniques to evaluate the integral in Equation 11 with simple lookups to Gaussian-prefiltered environment maps [McAllister et al. 2002].

The interpolation of the transport function is performed using barycentric interpolation of the Gaussian parameters. The nonlinear effect of these parameters enables high-quality rendering. At each of the three closest spatial samples $\mathbf{x}_{1,2,3}$ (triangle vertices), we lookup the Gaussians for the three closest directions $\mathbf{v}_{1,2,3}$. Corresponding Gaussians (see Section 4.2) at each vertex $\mathbf{x}_l$ are interpolated using the barycentric weights [Tan et al. 2005]. The resulting Gaussians at each $\mathbf{x}_{1,2,3}$ are then interpolated to yield the actual function $\tilde{T}_{\mathbf{x},\mathbf{v}}$ at position $\mathbf{x}$. It is trivial to rotate the incident lighting $L_i$, because the mean parameters $\mu_k$ are represented in a global coordinate frame. To rotate the lighting by $R_\theta$, we apply the inverse rotation $R_\theta^T$ to $\mu_k$, and use the resulting direction to index the lighting cubemap.

## 6 Results and Implementation

In this sections we describe our implementation and evaluate our results from a numerical, visual, and performance perspective.

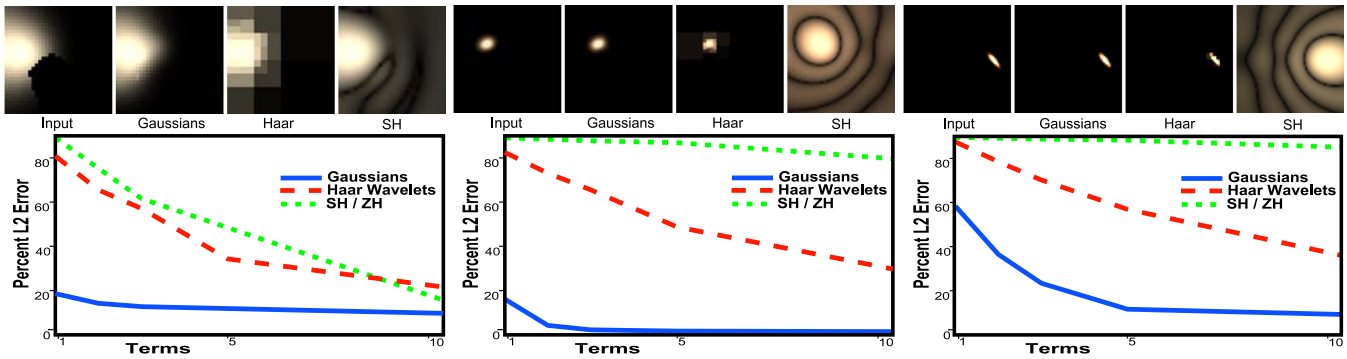



Figure 5: Visual and error analysis of three typical transport functions and their approximations. Top row of each subfigure shows from left to right: unapproximated data, Gaussian approximation using 10 Gaussian lobes, nonlinear wavelet approximation using the 70 largest coefficients, and $10^{\text{th}}$ order Spherical Harmonic approximation. Graphs shows percentage $L^2$ error as a function of number of Gaussians and the equivalent number of wavelet and spherical harmonic coefficients.

### 6.1 Precomputation

As noted earlier, it is necessary to compute the transport function at each vertex and for a set of fixed view directions, specified in a global coordinate frame. In practice we used 92 viewing directions, with on average 46 directions falling in the upper hemisphere of any given vertex. The view directions are defined in a global coordinate frame to simplify angular interpolation.

We decompose the transport operator into a diffuse (view-independent) component and a glossy (view-dependent) component (see Figure 4). For all the images appearing in this paper, as well as the accompanying video we represent the diffuse component using fifth-order linear spherical harmonics, however any other suitable technique could easily be used [Ng et al. 2003; Sloan et al. 2003].

We optimize the objective function described in Equation 6 and Section 4.2. In practice we use the MATLAB optimization toolbox routine *fmincon*. As nonlinear optimization is a compute intensive procedure, we used a cluster of 20 machines to distribute the work. Typical precomputation times ranged from one to three hours. It may be possible to significantly accelerate precomputation times by leveraging the sparseness of typical view-dependent transport data.

A single Gaussian is stored as seven floating point numbers: three for RGB weights, three for the mean, and one for the variance (see Equation 4). Although only approximately 46 of the 92 view directions are in the upper hemisphere of the surface normal, Gaussian parameters are computed and stored for all 92 view directions to allow smooth interpolation near the horizon boundary. The weights $w_k$ of views falling below the hemisphere are set to zero. The total storage for a model with $M$ vertices and a $N$ term Gaussian approximation is $92 \times 7 \times M \times N$ floating point numbers. For example, a model with 10K vertices using a one-term Gaussian approximation requires 25MB to store. Note that this is far from an optimal encoding of Gaussians, but it was chosen to simplify rendering. Clustering techniques [Sloan et al. 2003] and better encoding of zero-weighted Gaussians could further reduce storage costs.

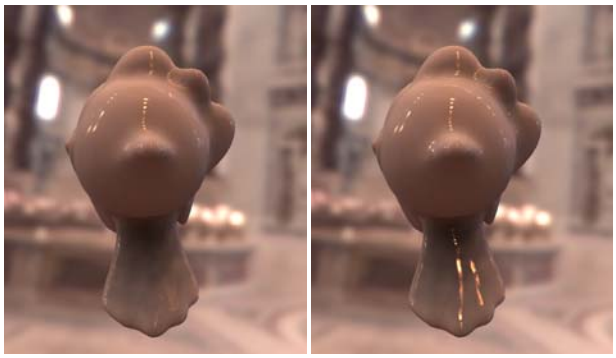
### 6.2 Rendering

Rendering is simple, with the majority of the computation occurring in vertex and fragment shaders. In practice, we have implemented the interpolation across view in a vertex shader, and the interpolation across triangles in a pixel shader. The integral of the interpolated Gaussians with the environment map is a simple mip-map look up to a prefiltered environment cubemap. The per-pixel interpolation of the Gaussian direction and variance is critical to achieve high visual quality. The CPU is only used to locate the data for the three nearest views of each vertex and pack it into vertex buffer objects to be sent to the GPU. In fact, even this stage could be accomplished on the GPU by adding several texture lookups to the vertex shaders. The main bottleneck of our implementation is the data transfer from the CPU to the GPU. Despite the unoptimized rendering code the glossy component can achieve frame rates of greater than 60Hz for modestly sized models (10–30K vertices) and the combined spherical harmonics and glossy component can be rendered at 30–40Hz. It takes approximately two seconds to preconvolve all levels of a $512 \times 512 \times 6$ (at the finest level) cubemap mip-map in software. Using a smaller input cubemap, or further optimization (e.g., GPU or SSE implementations) may be able to provide unconstrained per-frame dynamic lighting. We currently support arbitrary rotations of the input lighting in real-time.

### 6.3 Error Analysis and Comparisons

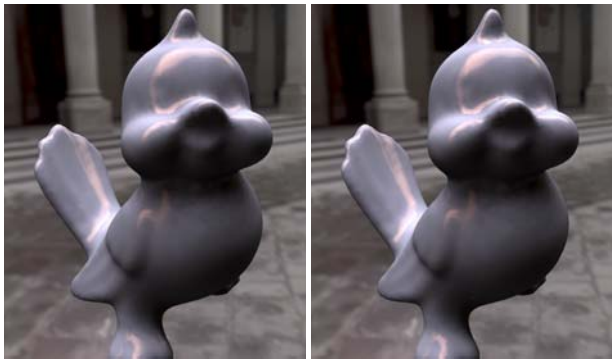
We demonstrate the expressive power of our model by calculating the relative $L^2$ error of approximated transport functions using our Gaussian representations, nonlinear Haar wavelet approximation [Ng et al. 2003] and spherical harmonics(SH). We choose not to compare directly with zonal harmonics (ZH) [Sloan et al. 2005], as ZH are a subset of SH. One might argue that the nonlinear fitting of ZH used by Sloan et al. [2005] may add more expressive power to ZH, however, it should be noted that ZH are band-limited in the same way as SH. Thus, high frequencies require higher-order ZH, which is prohibitively expensive during rendering.

Figure 5 shows relative $L^2$ error for three examples: A small symmetric glossy lobe, a large lobe with visibility, and a highly anisotropic, highly specular lobe. A fair comparison of representations as different as nonlinear Gaussians and truncated Haar wavelet is difficult. We have chosen to take into account the number of coefficients used in the representation. In practice, our Gaussians are represented using seven floating point values, we therefore decided to compare a one-Gaussian approximation with seven-coefficient truncated Haar and SH approximations. As noted above, this is not an optimal encoding of Gaussians. In addition, Haar encoding might require more RGB coefficients. The situation for ZH is a bit more complicated as they have the ability for multi-lobe approximations. However, since ZH are bandlimited, we feel it is sufficient (in terms of accuracy of the approximation) to compare only with SH. As such, the number of parameters was determined based on



(a) (b)

Figure 6: Visual comparison of the effect of visibility. Image (a) is rendered with our method. Image (b) is rendered using a prefiltering technique without visibility. A highlight that should be occluded is clearly visible on the tail of (b).



(a) (b)

Figure 7: Comparison of the bird model rendered with a single Gaussian component (a), and two Gaussian components (b). There are only minor differences in the visual appearance between the one-component and two-component approximations. We favor the use of a single Gaussian in practice.

a standard SH representation. Despite this generous allotment of coefficients, it is clear from Figure 5 that the Gaussian representation is superior both in terms of $L^2$ error and smoothness for such a small number of coefficients.

### 6.4 Visual Quality

The most important aspect of any rendering technique is the quality of the final images. The nonlinear effect of the Gaussian mean and variance parameters allow high-quality interpolation across views and spatial locations. The gain is similar to that of Phong interpolation, but additionally we encode visibility. Figure 6 shows a visual comparison between our method and a prefiltering method that omits visibility. Figure 2 demonstrates that fine highlight details are captured using a sparse sampling in both the view directions and mesh location. Figure 8 shows two example images rendered using our technique.

In our experience, one Gaussian per transport component is usually enough to obtain high subjective quality (see Figure 7 for a comparison). While the numerical gain is significant for subsequent Gaussians, the visual gain is not as high, partially because the visual complexity makes it harder to assess the complex reflection patterns. For practical purposes, we advocate the partition of light transport into multiple components when possible. The coherence terms in the optimization perform well, but we have sometimes observed Gaussian flipping, especially at the end of the mesh flood fill when neighboring vertices favor incompatible constraints. We believe that an additional relaxation step can improve this.

(a) (b)

Figure 8: Rendering of the Armadillo model (a) with $\sim$ 50K vertices and the Dane model (b) with 15K vertices. Full rendering frame rate for Armadillo is $\sim$ 10Hz. Dane renders at $\sim$ 40Hz. Both models are rendered using a single Gaussian lobe for the view-dependent component, and fifth-order spherical harmonics for the diffuse component.

## 7 Conclusions and Future Work

We have presented a real-time method for rendering static objects under distant all-frequency lighting, which captures view-dependent effects at a small storage cost. Our method is based on a nonlinear representation of light transport based on Gaussian mixture models. This representation presents several advantages: it is compact, can capture arbitrary bandwidth, and yields high-quality interpolation. We have presented an optimization approach to fit the Gaussian parameters to light transport data while achieving good correspondence across view direction and mesh location. The rendering phase of our technique involves simple shaders and achieves high quality and performance.

In the future, we would like to extend our system to dynamic scenes based on a similar representation. Furthermore, we are interested in removing the restriction to distant lighting, which raises a critical challenge to address the resulting high-dimensional space. In addition, we want to study the applicability of the EM approach to accelerate our parameter estimation. Further compression of the transport is possible, perhaps using clustering or vector quantization methods. In addition to reducing storage, we expect that compression could significantly improve rendering efficiency, and we hope to explore this in the future.

## Acknowledgments

The authors would like to thank the members of the MIT Computer Graphics Group for their help while preparing this document. In particular, we thank Eugene Hsu for providing valuable stylistic and editorial feedback, Eric Chan for being our GPU guru, and Jaakko Lehtinen for insightful comments and discussion. This work was supported by an NSF CAREER award 0447561 Transient Signal Processing for Realistic Imagery, a Ford Foundation predoctoral fellowship, a DFG Emmy-Noether fellowship, an INRIA équipe associée, and the MIT-France program.

## References


ASHIKHMIN, M., AND SHIRLEY, P. 2002. Steerable illumination textures. *ACM Transactions on Graphics 21*, 1 (Jan.), 1–19.

BLINN, J., AND NEWELL, M. 1976. Texture and reflection in computer generated images. *Communications of the ACM (SIGGRAPH '76 Proceedings) 19*, 10 (Oct.), 542–547.

CHUANG, Y.-Y., ZONGKER, D., HINDORFF, J., CURLESS, B., SALESIN, D., AND SZELISKI, R. 2000. Environment matting extensions: Towards higher accuracy and real-time capture. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, 121–130.

COLEMAN, T., AND LI, Y. 1996. An Interior, Trust Region Approach for Nonlinear Minimization Subject to Bounds. *SIAM Journal on Optimization 6*, 2, 418–445.

DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., AND SAGAR, M. 2000. Acquiring the reflectance field of a human face. In *Computer Graphics*, SIGGRAPH 2000 Proceedings, 145–156.

DEMPSTER, A., LAIRD, N., AND RUBIN, D. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society 39*, 1, 1–38.

DOBASHI, Y., KANEDA, K., NAKATANI, H., AND YAMASHITA, H. 1995. A Quick Rendering Method Using Basis Functions for Interactive Lighting Design. *Computer Graphics Forum 14*, 3 (August), 229–240.

DORSEY, J., SILLION, F., AND GREENBERG, D. 1991. Design and Simulation of Opera Lighting and Projection Effects. In *Proceedings SIGGRAPH*, 41–50.

FOURNIER, A. 1992. Normal distribution functions and multiple surfaces. In *Graphics Interface '92 Workshop on Local Illumination*, 45–52.

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

JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. AK Peters. ISBN: 1568811470.

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

KAUTZ, J. 2004. Hardware Lighting and Shading: A Survey. *Computers Graphics Forum 23*, 1 (Mar.), 85–112.

LAFORTUNE, E., FOO, S.-C., TORRANCE, K., AND GREENBERG, D. 1997. Non-linear approximation of reflectance functions. *Computer Graphics 31*, Annual Conference Series, 117–126.

LEHTINEN, J., AND KAUTZ, J. 2003. Matrix Radiance Transfer. In *Proceedings of the 2003 symposium on Interactive 3D graphics*, 59–64.

LIU, X., SLOAN, P.-P., SHUM, H.-Y., AND SNYDER, J. 2004. All-Frequency Precomputed Radiance Transfer for Glossy Objects. In *Proceedings Eurographics Symposium on Rendering 2004*, 337–344.

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

MATUSIK, W., PFISTER, H., ZIEGLER, R., NGAN, A., AND MCMILLAN, L. 2002. Acquisition and rendering of transparent and refractive objects. In *Rendering Techniques '02*, Springer, Wien.

MCALLISTER, D. K., LASTRA, A., AND HEIDRICH, W. 2002. Efficient rendering of spatial bi-directional reflectance distribution functions. In *HWWS '02: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 79–88.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-Frequency Shadows Using Non-linear Wavelet Lighting Approximation. *ACM Transactions on Graphics 22*, 3 (July), 376–381.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2004. Triple Product Wavelet Integrals for All-Frequency Relighting. *ACM Transactions on Graphics 23*, 3 (Aug.), 477–487.

NIMEROFF, J., SIMONCELLI, E., AND DORSEY, J. 1994. Efficient Re-rendering of Naturally Illuminated Environments. In *Fifth Eurographics Workshop on Rendering*, 359–373.

PRESS, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. 1992. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments. *ACM Transaction on Graphics 21*, 3 (July), 527–536.

SLOAN, P.-P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered Principal Components for Precomputed Radiance Transfer. *ACM Transactions on Graphics 22*, 3 (July), 382–391.

SLOAN, P.-P., LUNA, B., AND SNYDER, J. 2005. Local, Deformable Precomputed Radiance Transfer. *ACM Transactions Graphics 24*, 3 (Aug.).

TAN, P., LIN, S., QUAN, L., GUO, B., AND SHUM, H.-Y. 2005. Multiresolution Reflectance Filtering. In *Proceedings Eurographics Symposium on Rendering 2005*, 111–116.

TEO, P., SIMONCELLI, E., AND HEEGER, D. 1997. Efficient Linear Re-rendering for Interactive Lighting Design. Tech. Rep. STAN-CS-TN-97-60, Stanford University.

WANG, R., TRAN, J., AND LUEBKE, D. 2004. All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In *Proceedings Eurographics Symposium on Rendering 2004*, 345–354.

ZONGKER, D., WERNER, D., CURLESS, B., AND SALESIN, D. 1999. Environment matting and compositing. In *Computer Graphics*, SIGGRAPH 99 Proceedings, 205–214.