All-Frequency Interactive Relighting of Translucent Objects with Single and Multiple Scattering

Rui Wang John Tran David Luebke
University of Virginia *

Abstract

We present a technique, based on precomputed light transport, for interactive rendering of translucent objects under all-frequency environment maps. We consider the complete BSSRDF model proposed by Jensen et al. [2001], which includes both single and diffuse multiple scattering components. The challenge is how to efficiently precompute all-frequency light transport functions due to subsurface scattering. We apply the two-pass hierarchical technique by Jensen et al. [2002] in the space of non-linearly approximated transport vectors, which allows us to efficiently evaluate transport vectors due to diffuse multiple scattering. We then include an approximated single scattering term in the precomputation, which previous interactive systems have ignored. For an isotropic phase function, this approximation produces a diffuse transport vector per vertex, and is combined with the multiple scattering component. For a general phase function, we introduce a technique from BRDF rendering to factor the phase function using a separable decomposition to allow for view-dependent rendering. We show that our rendering results qualitatively match the appearance of translucent objects, achieving a high level of realism at interactive rates.

CR Categories: I.3.3 [Computer Graphics]: Picture/Image Generation; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism;

Keywords: Subsurface scattering, phase function, separable approximation, precomputed radiance transfer, Haar wavelets

1 Introduction

An accurate light transport model is essential for realistic image synthesis. Traditionally light scattering by materials is modeled by the BRDF (Bidirectional Reflectance Distribution Function), which assumes that light enters and exits the surface at the same point. Although this assumption is valid for metals, it is not valid for many translucent materials we commonly encounter in the natural world, such as marble, jade, wax, leaves, milk and human skin. Rendering these materials with the BRDF can create a hard, unconvincing appearance that overemphasizes small geometric details. On the appropriate scale most materials exhibit translucency, making them appear smooth and soft, blurring surface geometric details. This is due to light entering and being scattered within the object, a process known as subsurface scattering. Subsurface scattering has been simulated offline using a wide range of proposed methods for participating media, such as finite element methods [Rushmeier and Torrance 1987; Blasi et al. 1993], path tracing [Hanrahan and Krueger 1993; Lafortune and Willems 1996], photon mapping [Jensen and Christensen 1998; Dorsey et al. 1999] and diffusion approximation [Stam 1995]. Recent advances allow efficient simulation of subsurface scattering by the BSSRDF (Bidirectional Scattering Surface Reflectance Distribution Function) model, which relates outgoing radiance at a surface point to incident flux at all points on the surface. By assuming homogeneous media, Jensen et al. [2001] formulated the BSSRDF as the sum of single scattering and a diffuse dipole approximation for multiple scattering. Jensen et al. [2002] then presented a two-pass hierarchical integration technique to accelerate the computation of the multiple scattering component remarkably. They also experimentally validated that the importance of multiple scattering increases with the material albedo. Several recent papers [Lensch et al. 2002; Mertens et al. 2003; Hao and Varshney 2004] exploit this property and implement interactive systems for rendering multiple scattering. Unfortunately these techniques cannot efficiently handle illumination and shadows from an environment map, and they ignore single scattering.

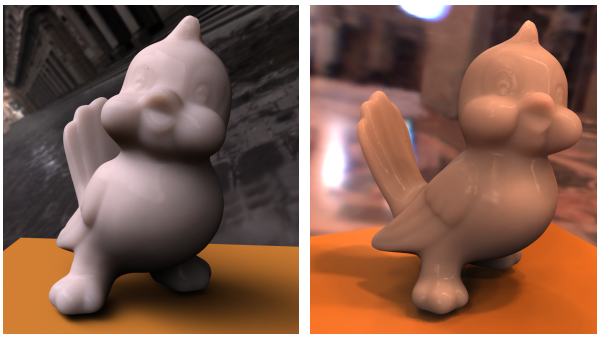


Figure 1: The bird model rendered with a BSSRDF that includes both single and multiple scattering under environment lighting. Note its translucent appearance and the all-frequency shadows.

Sloan et al. [2002] introduced precomputed radiance transfer (PRT) for real-time rendering with low-frequency environment lighting. They precompute for every vertex the radiance responses to a low-order (25D) spherical harmonics (SH) lighting basis, and store them as transport vectors. Relighting then reduces to the inner product of a light vector, represented in the same SH basis, with the precomputed transport vectors. Later, Sloan et al. [2003] included the diffuse multiple scattering component of the BSSRDF in their PRT framework. Single scattering is approximated using a glossy BRDF, but is not physically based. Due to linear approximation in an SH basis, their system is limited to low-frequency lighting. To improve the quality in all-frequency lighting environments, Ng et al. [2003] proposed non-linear approximation in a wavelet basis, and achieved interactive rates for diffuse BRDF rendering. This approach is later extended by [Wang et al. 2004] and [Liu et al. 2004] to render glossy BRDFs. To incorporate the BSSRDF, we face the challenge of deriving a compact formula for the light transport function that allows for efficient precomputation. A straightforward solution would be to precompute for every vertex the radiance responses to each lighting basis, then apply a non-linear

*e-mail: {rw2p, johntran, luebke}@cs.virginia.edu

wavelet approximation. This is feasible for a small number of SH lighting bases (25 in [Sloan et al. 2002]), however, it would incur an impractical amount of computation time for the large number of wavelet lighting bases (24,576 in [Ng et al. 2003]).

In this paper, we consider the complete BSSRDF model by Jensen et al., composed of both single and diffuse multiple scattering components. For multiple scattering, we apply their two-pass hierarchical technique [2002] in the space of transport vectors, which are non-linearly approximated using a wavelet basis. In the first pass, for selected surface sample points, we compute irradiance transport vectors, which describe irradiance values parameterized on the lighting environment. We compress these transport vectors using a non-linear wavelet approximation. In the second pass, we hierarchically integrate the precomputed irradiance transport vectors to compute a per-vertex multiple scattering transport vector. For single scattering, we use an approximated formula derived from [Jensen et al. 2001]. For an isotropic phase function, we evaluate a diffuse single scattering transport vector per vertex and combine it with the multiple scattering component. For a general phase function, we use a technique similar to [Wang et al. 2004] and [Liu et al. 2004]. Specifically, we factor the phase function using a separable decomposition [Kautz and McCool 1999] and keep K low-order terms, each consisting of a purely light-dependent part and a purely view-dependent part. We precompute K single scattering transport vectors per vertex, corresponding to each light-dependent part; the rendering algorithm then uses the view-dependent parts to determine vertex color. We show that our rendering results qualitatively match the appearance of translucent objects, achieving a high level of realism. To our knowledge, this is the first interactive system incorporating all-frequency environment lighting and subsurface scattering with both single and multiple scattering.

2 Background

In this section, we briefly review the BSSRDF formulation by Jensen et al. [2001] and their rapid two-pass hierarchical rendering technique [2002]. The formulation assumes a homogeneous participating medium, the properties of which are characterized by the absorption coefficient σ_a, the scattering coefficient σ_s and the phase function $p(\vec{\omega}_i, \vec{\omega}_o)$. The extinction coefficient is $\sigma_t = \sigma_a + \sigma_s$.

The BSSRDF describes subsurface scattering in such a medium by:

$$L_o(x_o, \vec{\omega}_o) = \int_A \int_{2\pi} S(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o)\, L(x_i, \vec{\omega}_i)\, (\vec{n}_i \cdot \vec{\omega}_i)\, d\vec{\omega}_i\, dA(x_i) \quad (1)$$

where L_o is the outgoing radiance at point x_o in direction $\vec{\omega}_o$, L is the incident radiance at point x_i in direction $\vec{\omega}_i$, and S is the BSSRDF. Jensen et al. [2001] define the BSSRDF as the sum of a single scattering term $S^{(1)}$ and a diffuse multiple scattering term S_d:

$$S(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o) = S^{(1)}(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o) + S_d(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o)$$

Using a dipole source approximation, they derive S_d as:

$$S_d(x_i, \vec{\omega}_i; x_o, \vec{\omega}_o) = \frac{1}{\pi} F_t(\eta, \vec{\omega}_i)\, R_d(||x_i - x_o||)\, F_t(\eta, \vec{\omega}_o) \quad (2)$$

where F_t is the Fresnel transmittance, η is the relative index of refraction, and R_d is the diffuse reflectance computed by:

$$R_d(r) = \frac{\alpha'}{4\pi} \left[z_r (\sigma_{tr} + \frac{1}{d_r}) \frac{e^{-\sigma_{tr} d_r}}{d_r^2} + z_v (\sigma_{tr} + \frac{1}{d_v}) \frac{e^{-\sigma_{tr} d_v}}{d_v^2} \right] \quad (3)$$

where $\sigma_s' = (1-g)\,\sigma_s$ and $\sigma_t' = \sigma_a + \sigma_s'$ are reduced scattering and extinction coefficients, $\alpha' = \sigma_s'/\sigma_t'$ is the reduced albedo, g is the mean cosine of the scattering angle, $\sigma_{tr} = \sqrt{3\,\sigma_a\,\sigma_t'}$ is the effective extinction coefficient, $d_r = \sqrt{r^2 + z_r^2}$ and $d_v = \sqrt{r^2 + r_v^2}$ are the distances from illumination point x_i to the dipole source, $r = ||x_o - x_i||$ is the distance between x_i and x_o, and $z_r = 1/\sigma_t'$ and $z_v = z_r(1 + 4A/3)$ are the distances from x_o to the dipole source. Here $A = (1 + F_{dr})/(1 - F_{dr})$, and F_{dr} is a diffuse Fresnel term approximated by $F_{dr} = -1.440/\eta^2 + 0.710/\eta + 0.668 + 0.0636\,\eta$.

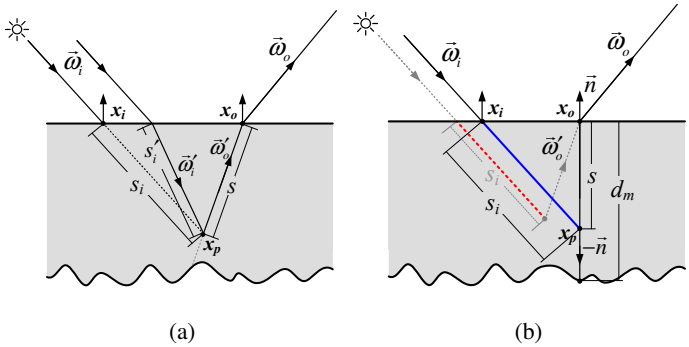


Figure 2: (a) Single scattering is computed by integrating over the refracted outgoing ray $\vec{\omega}_o'$. x_p is a sample point along the integration path. Incident path length s_i' is approximated from observed path length s_i by Eq 5. (b) We approximate single scattering by using the negative normal direction as the integration path. The blue solid line shows the approximated observed path length s_i, and the red dotted line is the true s_i. The approximation is more accurate as x_p gets closer to the surface, which also contributes more importance to the integral. d_m is the maximum distance along the path.

The diffusion approximation requires an expensive integration of light transport from all points on the surface. Jensen et al. [2002] introduced a two-pass approach to accelerate the integration process remarkably. The key idea is to decouple the computation of incident illumination from the evaluation of BSSRDF, making it possible to reuse illumination samples. Specifically, in the first pass they compute irradiance values for uniformly sampled points on the surface. These samples are stored in an octree, with each node storing clustered values of all its child nodes. In the second pass they traverse the octree by recursively subdividing nodes, evaluate the diffusion approximation using clustered node values, and employ a simple heuristic to decide the level to stop subdivision.

Single scattering is computed by the following equation in which the BSSRDF single scattering term $S^{(1)}$ is implicitly defined:

$$L_o^{(1)}(x_o, \vec{\omega}_o) = \sigma_s \int_{2\pi} \int_0^\infty F\, p(\vec{\omega}_i', \vec{\omega}_o')\, e^{-\sigma_t (s_i' + s)}\, L(x_i, \vec{\omega}_i)\, ds\, d\vec{\omega}_i \quad (4)$$

where $\vec{\omega}_i'$ and $\vec{\omega}_o'$ are the refracted incoming and outgoing directions, $F = F_t(\eta, \vec{\omega}_i) \cdot F_t(\eta, \vec{\omega}_o)$ is the combined Fresnel transmittance, s_i' and s are the scattering path lengths along $\vec{\omega}_i'$ and $\vec{\omega}_o'$, and p is a normalized phase function. When sampling the illumination, it is difficult to estimate s_i' accurately since that requires finding the point of refraction x_i for arbitrary geometry. In practice, if we assume that the surface at x_i is locally flat and illumination is distant, a good approximation of s_i' can be found [Jensen et al. 2001] by:

$$s_i' = s_i \frac{|\vec{\omega}_i \cdot \vec{n}_i|}{\sqrt{1 - (\frac{1}{\eta})^2 (1 - |\vec{\omega}_i \cdot \vec{n}_i|^2)}} \quad (5)$$

where s_i is the observed path length as if the incident ray is not refracted. The single scattering component (Eq 4) is derived from previous work by Hanrahan and Krueger [1993] and is computed by Monte Carlo integration along $\vec{\omega}_o'$. Figure 2(a) shows the scenario.

3 Algorithms and Implementation

In this section we derive the algorithms for precomputing transport functions due to the BSSRDF and present an accelerated GPU implementation. We assume a homogeneous medium illuminated by distant environment lighting and consider only direct illumination.

Diffuse Multiple Scattering Since lighting is distant, we have $L(x_i,\vec{\omega}_i) = L(\vec{\omega}_i)V(x_i,\vec{\omega}_i)$ where V is the visibility. Substituting Eq 2 into Eq 1, we can write down the outgoing radiance L_d due to diffuse multiple scattering as:

$$L_d(x_o,\vec{\omega}_o) = \frac{1}{\pi}F_t(\eta,\vec{\omega}_o)\int_{2\pi} L(\vec{\omega}_i)\,T_d(x_o,\vec{\omega}_i)\,d\vec{\omega}_i \quad (6)$$

$$T_d(x_o,\vec{\omega}_i) = \int_A R_d(||x_i-x_o||)\,E(x_i,\vec{\omega}_i)\,dA(x_i) \quad (7)$$

$$E(x_i,\vec{\omega}_i) = F_t(\eta,\vec{\omega}_i)\,V(x_i,\vec{\omega}_i)\,(\vec{n}_i\cdot\vec{\omega}_i) \quad (8)$$

where T_d is the multiple scattering transport function to be evaluated, E is the irradiance transport function at point of illumination x_i which describes light transport from the environment to x_i, and the diffuse reflectance R_d (defined in Eq 3) predicts the light transport from x_i to x_o, similar to *form factors* in radiosity methods. Using numerical cubature, we can evaluate the integral in Eq 6 as:

$$L_d(x_o,\vec{\omega}_o) = \frac{1}{\pi}F_t(\eta,\vec{\omega}_o)\sum_j T_d(x_o,\vec{\omega}_j)\,L(\vec{\omega}_j)$$

Now we can precompute T_d for every vertex and store it as a transport vector. Relighting then reduces to a simple dot product of T_d with the light vector L. Notice that the above relighting equation holds when L is expressed in any orthonormal basis, such as spherical harmonics or wavelets. This makes it possible to apply approximation techniques for efficient storage and fast rendering.

As in [Ng et al. 2003], we parameterize the distant lighting on a high resolution environment cubemap, and compress transport vectors T_d using non-linear wavelet approximation. To compute T_d efficiently, we apply the two-pass hierarchical technique by Jensen et al. [2002], but perform the computation in transport vector space. Specifically, in the first pass we compute irradiance transport functions E for a set of uniformly distributed surface sample points. We apply a wavelet transform on these transport functions, nonlinearly approximate them by keeping only a bounded number of the largest wavelet coefficients, then quantize and store the coefficients as compressed transport vectors. In the second pass, we build a kd-tree by clustering and summing the irradiance transport vectors computed in the first pass. To reduce memory consumption, we always keep a bounded number of wavelet coefficients at each node, and truncate the remaining. Next for each vertex we traverse the kd-tree in the same manner as Jensen et al. and evaluate transport vector T_d by accumulating clustered irradiance transport vectors on each node. The clustering and accumulating process is valid since the wavelet transform is a linear operator. As the node level goes up, some high-frequency components may be lost by truncation; however, by keeping an appropriate number of wavelet coefficients at each level, the artifacts are unnoticeable in our tests.

Single Scattering There are two difficulties in precomputing single scattering according to Eq 4. First, the integration path is the refracted outgoing direction, which is unknown at precomputation time. Second, a general phase function depends on both incident and view directions, making precomputation inseparable from rendering. To this end, we propose two approximations to make precomputation feasible. First, we always place sample points along the opposite direction of a vertex normal. By doing this, we have approximated the view-dependent integration path with a view-independent path known at precomputation time (see Figure 2(b)). This approximation changes the observed incident path length s_i. It is more accurate for sample points closer to the surface, which are also the more important samples for the integral due to the exponential attenuation in s. Second, we approximate the phase function using a separable decomposition technique introduced by [Kautz and McCool 1999], which has been used for BRDF rendering:

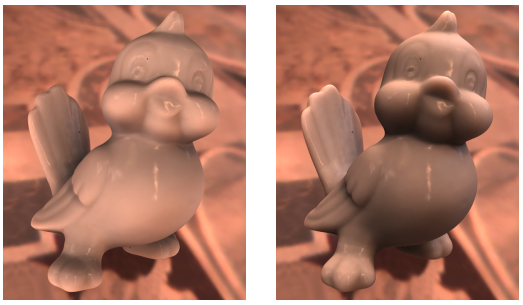


Figure 3: Rendering of single scattering for the bird model using phase function approximations. In both images a 4-term approximation of an HG phase function is applied: the left image with $g = 0.25$ (primarily forward scattering) and the right image with $g = -0.25$ (primarily backward scattering).

$$p(\vec{\omega}_i,\vec{\omega}_o) \approx \sum_{k=1}^{K} g_k(\vec{\omega}_i)\,h_k(\vec{\omega}_o) \quad (9)$$

where K is the number of approximation terms, g_k and h_k are the *light map* and *view map*, and are stored as textures to be indexed with incident and view directions respectively. A number of matrix factorization methods are available for the decomposition, from which we choose singular value decomposition (SVD). With this approximation, we can now handle phase functions in the same way as [Wang et al. 2004] and [Liu et al. 2004] handle glossy BRDFs in PRT. We substitute Eq 9 into Eq 4 and rearrange the terms:

$$L_o^{(1)}(x_o,\vec{\omega}_o) \approx F_t(\eta,\vec{\omega}_o)\sum_k h_k(\vec{\omega}_o{}')\int_{2\pi} L(\vec{\omega}_i)\,T_k^{(1)}(x_o,\vec{\omega}_i)\,d\vec{\omega}_i$$

$$T_k^{(1)}(x_o,\vec{\omega}_i) = \int_0^\infty \sigma_s\,T_p(x_p,\vec{\omega}_i)\,e^{-\sigma_t s}\,ds \quad (10)$$

$$T_p(x_p,\vec{\omega}_i) = g_k(\vec{\omega}_i{}')\,F_t(\eta,\vec{\omega}_i)\,e^{-\sigma_t s_i'}\,V(x_i,\vec{\omega}_i) \quad (11)$$

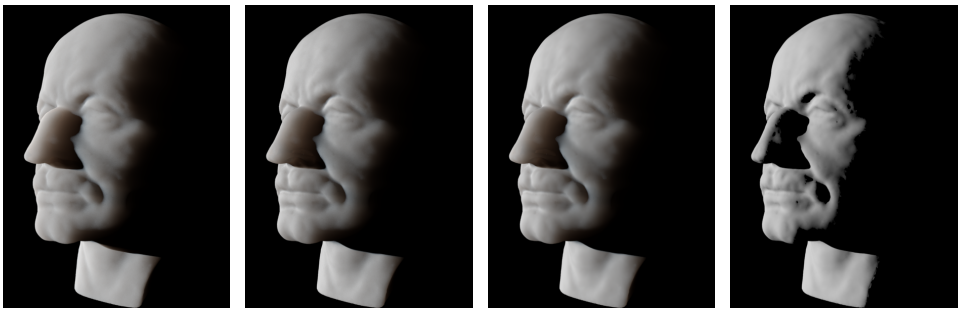
where $T_k^{(1)}$ is the k-th single scattering transport function, corresponding to the k-th phase function term. x_p is the sample point at distance s along the integration path, and T_p is the illumination transport function at x_p describing the amount of light transported to x_p from the environment but attenuated along the incident path. Notice that for an isotropic phase function, the single scattering transport function is in a diffuse form, and can therefore be combined with the multiple scattering component. Using numerical cubature, we write down the single scattering relighting equation as:

$$L_o^{(1)}(x_o,\vec{\omega}_o) = F_t(\eta,\vec{\omega}_o)\sum_k h_k(\vec{\omega}_o{}')\sum_j T_k^{(1)}(x_o,\vec{\omega}_j)\,L(\vec{\omega}_j)$$

Evaluation of $T_k^{(1)}$ can employ Monte Carlo techniques to integrate T_p over the sample path. For importance sampling, we pick a random distance along the sample path by $s(\xi) = -\ln(1-\lambda\,\xi)/|\sigma_t|$, where $\xi \in [0,1]$ is a uniformly distributed random number, $|\sigma_t|$ is the luminance of σ_t, $\lambda = 1-e^{-|\sigma_t| d_m}$ is the normalization term, and d_m is the integration upper limit, which is the maximum distance

Multiple Scattering | Single Scattering | Multiple and Single Scattering | Diffuse BRDF

Figure 4: Rendering of the bunny model with the components of a synthetic BSSRDF. We also show a BRDF rendering for comparison.



(a) (b) (c) (d)

Figure 5: Ray traced images of the single scattering term (Eq 4) for the head model using an HG phase function with $g = -0.25$. (a) is the true answer using the refracted outgoing ray as the integration path with 128 importance samples; (b) is our approximation using the negative normal direction as the integration path with 128 importance samples; (c) is similar to (b) but uses only 16 deterministic samples; (d) is rendered with a diffuse BRDF for comparison.

along the path. The probability distribution function for this importance sampling is $\gamma(s) = |\sigma_t| e^{-|\sigma_t| s} / \lambda$. In practice, since it takes many random samples to eliminate noise, we instead use a fixed number of N deterministic sample distances by taking $s = s(\xi_i)$, where $\xi_i = (i - 0.5)/N$ and $i = 1 \ldots N$.

Precomputation We parameterize the distant lighting on a $6 \times 32 \times 32$ cubemap. Irradiance and illumination transport functions are computed by rendering into OpenGL 16-bit floating point pbuffers. This eliminates the need to download intermediate data such as visibility maps to the CPU. For multiple scattering, we first evaluate irradiance transport functions for a set of evenly distributed surface sample points. At each point, we rasterize a simplified model onto the six cubemap faces and set the stencil buffer to one for every pixel drawn. The inverse of the stencil buffer, which now stores the visibility information, is then applied to generate the transport function according to Eq 8. For proper anti-aliasing, we use 2×2 super sampling in rasterization. We then down sample the data on the card and download the pbuffer to the CPU to perform the non-linear wavelet approximation. The results are quantized and stored as transport vectors similarly to [Ng et al. 2003]. Once irradiance sampling is completed, we apply the hierarchical integration to compute transport vectors T_d by the diffuse approximation.

For single scattering, we sample along the negative normal direction of each vertex. The integration upper limit d_m is precomputed using a ray tracer by shooting a ray along the negative normal direction and detecting the intersection. We use 16 deterministic samples to integrate over the path. At each sample point, we again render a simplified model into an OpenGL pbuffer with 2×2 super sampling. To correctly account for visibility, we set the stencil buffer to increment by one for each fragment generated. A stencil count above 1 thus means the incoming ray is occluded before entering the object. Also in this pass, we output vertex normal and depth for each fragment. These values are used in calculating ω_i' and s_i' for T_p in Eq 11. The single scattering transport functions are computed by summing the illumination transport functions of each sample point according to Eq 10. This is accomplished on the GPU with 16-bit floating point blending. If the phase function is isotropic, we then sum the single and multiple scattering transport vectors to produce a combined diffuse transport vector. For an arbitrary phase function, we first tabulate the phase function by sampling both incident and view directions on a $6 \times 16 \times 16$ cubemap, then perform a SVD on the tabulated data, keeping the first $K = 4$ terms and storing them as cubemap textures. Each term contains a pair of light map and view map. We apply the K light maps in precomputation as indicated by Eq 11 and produce K transport functions per vertex. In this case, each of the $K = 4$ transport functions is computed into its own surface of a pbuffer using multiple render targets, eliminating the need to recompute visibility for each term. We finally transfer downsampled data to the CPU to perform the final step of non-linear wavelet approximation. Computing the transport functions on the GPU is essential, providing us with a performance gain of an order of magnitude over an equivalent CPU implementation, which uses the GPU only for visibility sampling.

Rendering To render, we dynamically sample the environment map on a $6 \times 32 \times 32$ cubemap and apply a 2D Haar wavelet transform to produce the light vector. We compute the dot products of the light vector with the transport vectors on the CPU. For a diffuse transport vector, the results are scaled by the Fresnel transmittance to determine the final vertex colors. For a K-term phase function approximation, these results need to be further combined with texture lookups of the refracted view direction into the phase function view maps ($h_k(\vec{\omega}_o{}')$). The Fresnel term and $\vec{\omega}_o{}'$ are both computed in the fragment shader that also accesses the view map textures. In addition, since specular highlights play an important role in perceiving translucent objects, we use a pre-convolved environment map to add a specular component to the final rendering. Although the environment mapping technique ignores self-shadowing, it qualitatively simulates convincing appearance when combined with the translucent base color, which has taken into account self-shadowing.

(a) (b) (c) (d) (e) (f)

Figure 6: From (a) to (d) we show renderings of single scattering using a 1-term, 4-term, 8-term and 16-term approximation for the HG phase function with $g = -0.4$. (e) is rendered with the analytic phase function for comparison. (f) plots the decay of the RMS error of our approximation as the number of terms increase. Each curve denotes an HG phase function with a different parameter g.

4 Results and Discussion

In this section we show our experimental results and verify our single scattering approximations. Timings are recorded on an Intel Pentium 4 2.53 GHz computer with 1 GB memory and an NVIDIA GeForce 6800 GT graphics card. Our programs are compiled using Intel Compiler version 8.0.

Figure 1 shows the bird model rendered with a synthetic BSSRDF ($\sigma_s = [0.75, 0.85, 1.00]$, $\sigma_a = [0.02, 0.04, 0.07]$) that includes both multiple scattering and single scattering with an isotropic phase function. To demonstrate our technique in all-frequency lighting environments, we precompute this model using a $6 \times 64 \times 64$ cubemap resolution and keep 256 ($\sim 1\%$) wavelet coefficients. Note the high and low frequency shadows on the diffuse floor. This model contains $60K$ vertices including the floor. It requires 45 minutes total precomputation time, and we can relight the model at 6 fps. For all other tests, we use a smaller cubemap resolution of $6 \times 32 \times 32$, and keep 128 ($\sim 2\%$) wavelet coefficients. Subsurface scattering tends to blur the light across shadow boundaries and soften the overall look of an object. Therefore compared with the BRDF, the BSSRDF transport functions are relatively low-frequency, and a lower cubemap resolution thus suffices. A good reference for accuracy analysis of the non-linear wavelet approximation can be found in [Ng et al. 2003]. The precomputation time, storage size and relighting speed for our test models are listed in Table 1. View-dependent rendering is maintained at real-time rates and are not listed.

Figure 4 and Figure 7 show renderings of the BSSRDF components and compare them with a diffuse BRDF rendering. Figure 8 shows the BSSRDF rendering of the dragon model with a Perlin style marble texture. Note the translucent appearance of the objects and how the single scattering component adds a solid yet translucent look to the models. In all these examples we use synthetic BSSRDF parameters so that the contributions from the single and multiple scattering components are about the same, and the single scattering components are all computed with an isotropic phase function.

Single Scattering Approximation To verify our approximations with single scattering, we render the single scattering component for the head model using a Monte Carlo ray tracer according to Eq 4. We use the BSSRDF parameters of a measured marble material from [Jensen et al. 2001] and apply a Henyey-Greenstein (HG) phase function with $g = -0.25$ (primarily backward scattering). Figure 5 shows the comparison of the ray traced images and the model is illuminated with a single distant light. Image (c) is rendered with our approximation for the integration path and the deterministic sampling method, which qualitatively matches the true answer in (a) yet is significantly faster. Note that the BRDF rendering in (d) is distinctly different from the other three.

	Bird	Buddha	Bunny	Dragon
No. of Vertices	31 K	56 K	72 K	78 K
Simp. Vis.	12 K	20 K	15 K	24 K
M.S. Sample Pts.	150 K	250 K	300 K	300 K
M.S. pass 1 P.T.	6 min	20 min	14 min	26 min
M.S. pass 2 P.T.	5 min	13 min	12 min	16 min
S.S. iso. P.T.	14 min	67 min	42 min	102 min
S.S. 4-term P.T.	17 min	73 min	48 min	108 min
Diff. Size	20 MB	37 MB	47 MB	51 MB
Diff. Light	25 fps	14 fps	12 fps	10 fps
4-term Size	79 MB	148 MB	188 MB	204 MB
4-term Light	6.5 fps	3.5 fps	3 fps	2.5 fps

Table 1: Precomputation and rendering profiles for our test models. Each column lists the model size, the size of the simplified model for visibility sampling; the number of sample points and the two-pass precomputation time for multiple scattering; precomputation time for single scattering, with an isotropic phase function and a 4-term separable phase function approximation; precomputed data size and relighting speed for a diffuse and 4-term transport vectors.

To analyze our phase function approximation, we again use a ray tracer to render the single scattering component for the head model, with an increasing number of phase function approximation terms K. In Figure 6 (a)-(e) we compare the results for an HG phase function with $g = -0.4$. We also plot the decay of RMS errors of our approximation for several HG phase functions. As the parameter g increases, the phase function becomes more directional and sharply shaped, therefore requiring more terms to approximate accurately. With K=4 we are limited to a low-order approximation of the phase function. This is similar to the BRDF where more specular BRDFs require more terms to approximate accurately. Finally we show two renderings of the bird model under environment lighting in Figure 3. They are rendered using the HG phase function with $g = 0.25$ (primarily forward scattering) and $g = -0.25$ (primarily backward scattering) respectively. Note their different appearances.

5 Conclusion and Future Work

Translucency is an attractive and important effect, but it is also challenging to simulate. In this paper we have presented a method, based on precomputed light transport, for interactive rendering of translucent objects under all-frequency environment maps. We incorporate the complete BSSRDF model in our precomputation, including both single and diffuse multiple scattering components. For single scattering, we propose several approximations to allow for

Figure 7: From left to right, the buddha model is rendered with the BSSRDF multiple scattering component, combined multiple and single scattering components, and a diffuse BRDF for comparison.

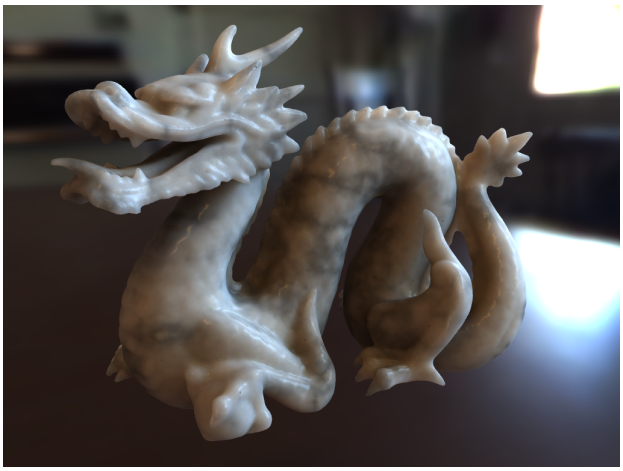


Figure 8: The dragon model rendered with the combined BSSRDF single and multiple scattering components. A Perlin style marble texture is applied during the final rendering.

view-dependent rendering. This improves on previous interactive systems, which have ignored the single scattering component. In the future we plan to accelerate our precomputation for single scattering by reusing illumination samples. The current system considers only direct illumination, and we would like to include global illumination from outside the medium. We also plan to incorporate multi-layered models into our system, which can increase accuracy and realism for certain materials such as leaves and skin.

Acknowledgements The authors would like to thank Paul Debevec for the light probe images, Cliff Woolley for providing the simplified models, and Matt Pharr and Greg Humphreys for the *pbrt* ray tracing system. The 3D models are courtesy of the Stanford University Computer Graphics Lab. and the Suggestive Contour Gallery (http://www.cs.princeton.edu/gfx/proj/sugcon/models/)

References

BLASI, P., SAËC, B. L., AND SCHLICK, C. 1993. A rendering algorithm for discrete volume density objects. *Computer Graphics Forum 12*, 3, 201–210.

DORSEY, J., EDELMAN, A., JENSEN, H. W., LEGAKIS, J., AND PEDERSEN, H. K. 1999. Modeling and rendering of weathered stone. In *Proc. of SIGGRAPH '99*, 225–234.

HANRAHAN, P., AND KRUEGER, W. 1993. Reflection from layered surfaces due to subsurface scattering. In *Proc. of SIGGRAPH '93*, 165–174.

HAO, X., AND VARSHNEY, A. 2004. Real-time rendering of translucent meshes. *ACM Trans. Graph. 23*, 2, 120–142.

JENSEN, H. W., AND BUHLER, J. 2002. A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph. 21*, 3, 576–581.

JENSEN, H. W., AND CHRISTENSEN, P. H. 1998. Efficient simulation of light transport in scences with participating media using photon maps. In *Proc. of SIGGRAPH '98*, 311–320.

JENSEN, H. W., MARSCHNER, S. R., LEVOY, M., AND HANRAHAN, P. 2001. A practical model for subsurface light transport. In *Proc. of SIGGRAPH '01*, 511–518.

KAUTZ, J., AND MCCOOL, M. D. 1999. Interactive rendering with arbitrary brdfs using separable approximations. In *Proc. of the 10th Eurographics Rendering Workshop*, 281–292.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1996. Rendering participating media with bidirectional path tracing. In *Proc. of the 7th Eurographics Rendering Workshop*, 91–100.

LENSCH, H. P. A., GOESELE, M., BEKAERT, P., KAUTZ, J., MAGNOR, M. A., LANG, J., AND SEIDEL, H.-P. 2002. Interactive rendering of translucent objects. In *Proc. of the 10th Pacific Graphics*, 214–224.

LIU, X., SLOAN, P., SHUM, H.-Y., AND SNYDER, J. 2004. All-Frequency Precomputed Radiance Transfer for Glossy Objects. In *Proc. of the 15th Eurographics Symposium on Rendering*, 337–344.

MERTENS, T., KAUTZ, J., BEKAERT, P., SEIDELZ, H.-P., AND REETH, F. V. 2003. Interactive rendering of translucent deformable objects. In *Proc. of the 14th Eurographics Symposium on Rendering*, 130–140.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. 22*, 3, 376–381.

RUSHMEIER, H. E., AND TORRANCE, K. E. 1987. The zonal method for calculating light intensities in the presence of a participating medium. In *Proc. of SIGGRAPH '87*, 293–302.

SLOAN, P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Trans. Graph.*, vol. 21, 527–536.

SLOAN, P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph.*, 382–391.

STAM, J. 1995. Multiple scattering as a diffusion process. In *Proc. of the 6th Eurographics Rendering Workshop*, 41–50.

WANG, R., TRAN, J., AND LUEBKE, D. 2004. All-Frequency Relighting of Non-Diffuse Objects using Separable BRDF Approximation. In *Proc. of the 15th Eurographics Symposium on Rendering*, 345–354.