EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Bidirectional Importance Sampling for Unstructured Direct Illumination

Rui Wang[1] and Oskar Åkerlund[2]

[1]University of Massachusetts Amherst
[2]Linköping University/University of Massachusetts Amherst

Abstract

*Recent research in bidirectional importance sampling has focused primarily on structured illumination sources such as distant environment maps, while unstructured illumination has received little attention. In this paper, we present a method for bidirectional importance sampling of unstructured illumination, allowing us to use the same method for sampling both distant as well as local/indirect sources. Building upon recent work in [WFA*05], we model complex illumination as a large set of point lights. The subsequent sampling process draws samples only from this point set. We start by constructing a piecewise constant approximation for the lighting using an illumination cut [CPWAP08]. We show that this cut can be used directly for illumination importance sampling. We then use BRDF importance sampling followed by sample counting to update the cut, resulting in a bidirectional distribution that closely approximates the product of the illumination and BRDF. Drawing visibility samples from this new distribution significantly reduces the sampling variance. As a main advance over previous work, our method allows for unstructured sources, including arbitrary local direct lighting and one-bounce of indirect lighting.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

As described in [Kaj86], the computation of rendering involves estimating a hemispherical integral of the lighting, visibility, and surface BRDFs. Numerical simulation of the integral often uses Monte Carlo sampling [Vea98]. It is well-known that the efficiency of Monte Carlo methods can be dramatically improved by using a proper importance sampling strategy. Often this requires drawing samples from a known distribution that is closely correlated with the integrand. Since the integrand in rendering involves the product of several terms, drawing samples simply from one of the terms typically produces poor results when the other terms contain high-frequency changes.

Recently, researchers have proposed several *bidirectional importance sampling* methods [BGH05] that draw samples according to the product distribution of the illumination function and surface BRDF. These methods incorporate important high-frequency changes in both terms, therefore can drastically reduce the sampling variance and improve the overall image quality with the same rendering time.

Most existing bidirectional methods, however, focus on structured light sources that can be naturally parameterized as 2D images. One popular choice is distant illumination represented as HDR environment maps. In this case, the illumination radiance along any arbitrary ray can be immediately obtained through texture access, making it feasible to start from the BRDF distribution and efficiently evaluate its joint distribution with the lighting [BGH05]. In addition, image-based sources enable the use of basis projection methods such as wavelets to separately approximate the lighting and BRDF in advance, then construct a combined sampling distribution function on the fly [CJAMJ05].

Unfortunately these approaches do not easily extend to unstructured light sources which may not have a natural parametrization. Examples includes arbitrary local direct lighting, and indirect lighting where the source illumination comes from the scene itself. The main difficulty is that due to the unstructured nature of the source, it is no longer trivial to obtain the radiance along any arbitrary ray. In addition, approaches based on wavelet projection require the source to be parameterized as 2D images. This requirement is non-trivial to satisfy for unstructured lights.

In this paper, we present a simple and efficient method for bidirectional importance sampling with unstructured direct illumination. As suggested by recent work [WFA*05, HPB07, AUW07], we assume that complex illumination can be sufficiently sampled at many point lights distributed over the source. The subsequent sampling process then draws samples only from this point set. We start by constructing piecewise constant clusters of the lighting using the illumination cut approach proposed by [CPWAP08]. We show that this cut can be used directly for illumination importance sampling. Next, for each pixel to be shaded, we apply BRDF importance sampling to cast a small number of secondary rays. We keep track of which illumination cluster(s) each ray hits by using a BVH, and then use the total number of BRDF samples fallen into each cluster to adjust the importance value stored at that cluster. The updated cut now closely approximates the product distribution of the illumination and BRDF. Finally, we draw visibility samples from this new distribution and complete the shading.

As a major advance over previous work, our method is no longer restricted to structured sources. This flexibility allows us to use the same method for sampling both distant sources as well as local sources, such as projected lighting and one-bounce of indirect lighting. In addition, our method shares a number of advantages with previous techniques [BGH05, CAM08b]: 1) it requires no precomputed BRDF or visibility data, thus is suitable for handling changes of the scene; 2) it only incurs a moderate overhead for constructing the bidirectional sampling function; 3) it is easy to implement on top of an existing ray tracer.

2. Related Work

Importance sampling reduces sampling variance by using a selected distribution – called the importance function – that is closely correlated with the integrand itself. High-energy regions in the integrand contribute more to the result, therefore require higher sampling density to improve efficiency.

Importance sampling from environment maps. Many algorithms construct importance functions based on a single term in the rendering equation, such as the illumination function. In photorealistic rendering, illumination is often modeled as distant HDR environment maps [DM97]. Much previous research in this direction has focused on distributing samples according to the energy distribution in the environment map, such as by using stratified sampling [CD], hierarchical sampling [Deb05], structured importance sampling [ARBJ03], fast blue noise sampling [ODJ04], and interleaved sampling [KK03]. As the importance of BRDF is ignored, these methods provide poor efficiency in the presence of highly glossy materials.

Importance sampling from BRDFs. BRDF importance sampling has also been studied extensively. Simple BRDF models, such as the Phong, have analytic integrals and can be sampled at high efficiency. More complex BRDFs can be sampled by using an appropriate distribution that looks like the BRDF itself. For measured BRDFs, [LRR04] introduced a general sampling method based on factored BRDF representations. These sampling methods rely purely on the BRDF, therefore face efficiency problems when the lighting contains high-frequency changes.

Bidirectional importance sampling Recent work has focused on sampling the product distribution of several functions involved in the rendering equation. Veach [Vea98] proposed a multiple importance sampling (MIS) strategy that draws samples from the illumination and BRDF separately, and then combines the results to reduce the overall sampling variance. A recent work by Burke et al. [BGH05] introduced *bidirectional importance sampling*, which samples the product distribution of the illumination and BRDF. They proposed an approach based on sampling importance resampling (SIR), which is also explored by [TCE05].

SIR works by drawing an initial, relatively large set of samples from one distribution, then evaluating these samples at a second distribution, and drawing a smaller set of final samples accordingly. Our method differs from SIR in that our final samples are not drawn from the initial, therefore our method is less biased towards the distribution where these initial samples are selected from. See Section 3.4 for a more detailed discussion.

Wavelet importance sampling [CJAMJ05] uses nonlinear wavelet approximation and triple product wavelet integrals [NRH04] to rapidly construct bidirectional importance functions on the fly. This approach requires precomputed BRDF data. [CETC06] eliminated the precomputation requirement by splitting environment maps recursively based on the peaks of a dynamic BRDF. Later, [CAM08b] presented an improved approach that directly computes a wavelet representation for the BRDF using a quadtree. In addition, they introduced a faster algorithm for computing the product of two wavelet functions.

A main limitation with these methods is that they are restricted to structured illumination sources that are parameterized as 2D images. This presents difficulties in dealing with unstructured sources such as indirect lighting and arbitrary local direct lighting. The fundamental difference in our method is that we model complex illumination as a large, unstructured set of point lights, and the subsequent sampling process draws samples only from this point set. This approach unifies the treatment of both structured and unstructured light sources, providing the same sampling efficiency but much improved flexibility.

A number of recent efforts [GH06, CAM08a, DWF06] explore visibility importance sampling to further improve the efficiency of Monte Carlo sampling. These methods typically exploit the spatial/temporal coherence in visibility to provide an approximation of the visibility function. In general, visibility importance sampling is less frequently used due to the high cost of sampling visibility on the fly.

[Jen95] introduced a method that uses the photon map to efficiently estimate the incident radiance arriving at a point, where the radiance is represented on a low-resolution map transformed by an invertible BRDF distribution. Since the incident radiance has already included visibility, this method properly accounts for all the important factors involved in shading. [SL06] extended this approach to more general materials by eliminating the need for an invertible BRDF. [HP02, Pha] also exploit the photon map to estimate illumination importance, but combine it with MIS to account for BRDF importance. The main limitation of these methods is that the use of a small number of photons (typically 50) is reasonable for low-frequency lights but insufficient for detailed high-frequency sources, such as image-based lighting.

Illumination from many lights. To reduce the rendering cost, a common technique is to convert complex illumination to a large number of point sources. Standard algorithms using this approach entail a linear cost with the number of lights. Recently, Walter et al. [WFA*05] introduced Lightcuts – a scalable sublinear solution for handling many lights using hierarchical clustering. This method exploits the illumination coherence using *cuts* that represents piecewise constant approximations. Our method is built directly upon theirs, but we focus on importance sampling, and provide a more efficient way for handling arbitrary BRDFs. The matrix row-column sampling algorithm by [HPB07] exploits the GPU to sample and cluster many lights, providing improved rendering speed. However, by using the same clustering of lights for the entire scene, their method is biased and is not suitable for glossy BRDFs.

[AUW07] explore the idea of cuts in precomputed visibility, achieving interactive relighting with dynamic BRDFs. More recently, [CPWAP08] proposed a fast algorithm for combining multiple cuts on the GPU. These methods require significant precomputed visibility data, therefore the rendering quality is limited by the precomputation accuracy and the vertex sampling rate. Our method differs from theirs in that we use bidirectional importance sampling to cast visibility samples per pixel on the fly, eliminating the need for any precomputed data. As a result, ours gives an unbiased solution, and the rendering quality is not restricted by precomputation. Furthermore, we provide an improved method for evaluating the BRDF average per cluster, resulting in more robust estimation for highly glossy materials.

3. Algorithm Overview

Assumptions. We make two assumptions that are similar to previous work in [WFA*05, HPB07]. First, we assume that illumination can be modeled as many diffuse (isotropically radiating) point lights distributed over the source. These points may be infinitely far away, representing distant illumination. As the number of points is sufficiently large, subsequent sampling only needs to draw samples within this point set. Second, we only consider direct illumination from the point lights. This covers both direct illumination received from distant or local sources, and one final bounce of indirect illumination received from the scene surface. In the latter case, we assume that the direct lighting is simple enough to be computed on the GPU, e.g. using shadow mapping.

3.1. Importance Sampling from a Set of Point Lights

We use $\mathcal{S}$ to denote the set of point lights. In the case of local illumination, each point in the set is associated with a position, a small surface area, and a normal. In the case of environment illumination, each point is infinitely far away, and is associated with a direction and a small solid angle.

The reflected radiance B caused by direct illumination from $\mathcal{S}$ to a surface point x_o is computed by:

$$B(x_o,\omega) = \sum_{\mathcal{S}} L(x_i)\, f_r(x_i \to x_o,\omega)\, V(x_i)\, G(x_i)\, \cos\theta_i \quad (1)$$

where ω is the viewing direction, $x_i \in \mathcal{S}$ is a point light, L is the source radiance, f_r is the surface BRDF, and V is the binary visibility function. Here $G(x_i)$ is the solid angle subtended by x_i at x_o, and θ_i is the incident angle.

To simplify the notation, we combine both $\cos\theta_i$ and G into f_r, and then focus on a fixed surface point at a fixed viewing direction, resulting in:

$$B = \sum_{\mathcal{S}} L(x_i)\, f_r(x_i)\, V(x_i) \quad (2)$$

Directly evaluating this summation is impractical as $\mathcal{S}$ typically contains a large number ($\geq$ 32K) of points. Instead, we can use Monte Carlo importance sampling to improve the efficiency of this computation. An unbiased Monte Carlo estimator of Eq 2 is given by:

$$B \approx \frac{1}{N} \sum_{s=1}^{N} \frac{L(x_s)\, f_r(x_s)\, V(x_s)}{p(x_s)} \quad (3)$$

where N is the number of Monte Carlo samples, p is a probability density function (PDF), and $x_s \in \mathcal{S}$ is a point light selected from $\mathcal{S}$ according to distribution p. A naive example for p is a uniform distribution: $p = \frac{1}{|\mathcal{S}|}$, with $|\mathcal{S}|$ being the total number of point lights. In this case, a point x_s is uniformly randomly selected from $\mathcal{S}$ and used to estimate Eq 3. However, this often results in high sampling variance.

The theory in Monte Carlo methods says that sampling variance can be dramatically reduced by using a proper distribution p that is closely correlated with the integrand. Take the illumination function L as an example: once L is known, we can use the luminance of L to define a PDF, such that $p(x_i) \sim L(x_i)$. This can be achieved by first building a discrete CDF (cumulative distribution function) from L, and using the inverted CDF to draw samples. Refer to [PH04] for details of implementation.

3.2. Importance Sampling from an Illumination Cut

As the number of point lights $|\mathcal{S}|$ is large, directly sampling from L is still quite expensive. Therefore it is necessary to exploit the coherence in L and reduce it to a lower-dimensional vector. Although a large body of previous work has studied using Haar wavelets to approximate L, these methods are typically limited to distant environment lighting, some additionally requiring precomputed BRDF data.

Representation using cut. An alternative representation, called *cuts* [WFA*05], has been recently presented and studied by several researchers [WABG06, AUW07, AWB08]. A cut approximates a function defined over $\mathcal{S}$ using piecewise constants. For example, an illumination cut can be computed for L by following the algorithm in [CPWAP08]. Specifically, they approximate L by partitioning $\mathcal{S}$ into a small number of clusters, such that the luminance of L within each cluster is coherent and hence all cluster members simply take the cluster's average value:

$$\langle l_k \rangle = \frac{1}{|\mathcal{C}_k|} \sum L(\mathbf{x}_i), \quad \mathbf{x}_i \in \mathcal{C}_k \tag{4}$$

where $\langle\cdot\rangle$ denotes the average, and $\mathcal{C}_k$ denotes a cluster. The geometric center of the cluster $\langle \mathbf{x}_k \rangle = \frac{1}{|\mathcal{C}_k|} \sum \mathbf{x}_i$ is computed and stored as a representative light of each cluster.

To create the clusters, a binary light tree is built from the point set $\mathcal{S}$ using a geometric distance metric. As shown in Fig 1, individual lights are placed at the leaf nodes and interior nodes represent clusters. During the building process, we compute the bounding box of the clustered lights at each tree node. This information will be used in the bidirectional sampling step. Next, the source radiance L is sampled at all leaf nodes, and the values are aggregated to interior nodes. At each node we compute the cluster average $\langle l_k \rangle$, and the variance $\mathrm{var}(l_k)$ which corresponds to the L^2 error resulting from the piecewise constant approximation. Finally, a cut through the tree is selected such that each node on the cut represents a disjoint cluster. This results in a piecewise constant approximation of L.

The selection of cut starts at the root node of the tree, followed by recursive subdivision. At each subdivision step, a node on the current cut with the highest L^2 approximation error will be selected and replaced with its two children nodes. The recursive process stops when the L^2 error of each node on the cut falls below a predefined threshold σ^2:

$$|\mathcal{C}_k|\,\mathrm{var}(l_k) \le \sigma^2, \quad \forall k \tag{5}$$

Illumination importance sampling using cut. For convenience, we use vector $\overrightarrow{L_c}$ to denote the illumination cut. Each element of $\overrightarrow{L_c}$ corresponds to a node (cluster) on the cut. The size of the cut typically varies between $300 \sim 1000$ in

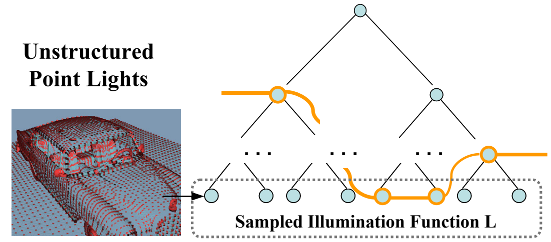


Figure 1: *A binary light tree and cut. Leaves represent individual point lights, and interior nodes clustered lights.*

our experiments. Note that $\overrightarrow{L_c}$ is simply a lower-dimensional version of the original L, therefore we can use it directly for efficient importance sampling. This is achieved in two steps. In the first step, we treat $\overrightarrow{L_c}$ as a discrete PDF, and use the inversion method [PH04] to draw a cluster from it. Each cluster has a probability value that is proportional to its total luminance: $|\mathcal{C}_k|\langle l_k \rangle$.

In the second step, for the cluster that we have picked above, we draw a sample by uniformly sampling the cluster's leaf nodes. As we assume each cluster represents a piecewise constant, all its children nodes have the same importance value. Therefore it suffices to pick a uniformly random sample among all the children nodes. The accuracy of this approach depends on how well the cut approximates the lighting. It is important that we compute the cut by minimizing the per-cluster error, as in Eq 5. This way, we can avoid getting high variance across different clusters.

The overall probability of picking a sample x_s is the joint probability of the two steps:

$$p(x_s) = p(x_s \in \mathcal{C}_k) \cdot p(x_s | x_s \in \mathcal{C}_k) \sim |\mathcal{C}_k|\langle l_k \rangle \cdot \frac{1}{|\mathcal{C}_k|} = \langle l_k \rangle$$

Note that since we use an unbiased estimator, the illumination cut does not introduce bias: it only affects the efficiency and convergence speed of our method.

3.3. Bidirectional Importance Sampling

We now extend the approach in the previous section to bidirectional importance sampling, where the sampling distribution p approximates the product of L and f_r. Our basic approach is to start with the illumination cut, then modify the importance value stored at each cut node by multiplying it with an estimation of the BRDF, thus incorporating BRDF importance. Ideally this would require computing the average BRDF value $\langle \rho_k \rangle$ for each cluster. Unfortunately, as the shape of each cluster is not clearly defined, accurately estimating $\langle \rho_k \rangle$ turns out to be a non-trivial task.

Single point sampling. One possibility is to use a single sample at each cluster to approximate the average BRDF value. This is essentially the approach taken by [AUW07], where they use each cluster's representative light to evaluate the BRDF, and use the result to directly substitute $\langle \rho_k \rangle$. We found that this idea works quite well in many cases, but it presents two major limitations. First, it entails a BRDF sampling cost of $O(|\overrightarrow{L_c}|)$, which is linear to the illumination cut size. Because BRDF sampling is expensive, as the illumination cut grows larger, this approach will quickly become too costly to be useful. Second, as the BRDF may contain very high frequencies relative to the cluster size, a single point sampling can result in severe aliasing artifacts, such as missing important peaks of the BRDF. This could be resolved by using more samples per cluster to evaluate the BRDF, but at the cost of increased computation time.

BRDF importance sampling. To address these problems, we propose an alternative approach that makes use of BRDF importance sampling followed by sample counting to lower the BRDF sampling requirement and improve efficiency. To start, we use standard BRDF importance sampling to generate a small number (N_ρ) of secondary rays. For example, the Phong specular BRDF can be sampled via

$$(\theta, \phi) = \left(\arccos(\sqrt[n+1]{\xi_1}), 2\pi\xi_2 \right)$$

where ξ_1 and ξ_2 are two independent uniform random variables, n is the Phong exponent parameter, and (θ, ϕ) are the spherical coordinates oriented at the local surface normal.

Since the BRDF samples are generated according to the BRDF distribution, we can think of all samples as reflected photons that carry equal amount of energy. Therefore, by counting the number of BRDF samples that fall into each cluster, we can reliably estimate the density of BRDF samples and hence the average BRDF value at each cluster.

In order to keep track of which cluster(s) a BRDF sample hits, we use a Bounding Volume Hierarchy (BVH) of the point lights. As shown in Fig 1, a BVH is naturally imposed by the structure of the light tree. This is done by computing a bounding box at each tree node during the tree building process. When a BRDF sample ray is generated, we perform ray-box intersection tests recursively, starting from the root node. The recursion stops whenever it has come to a cluster node on the cut, or if the intersection test has failed. See Figure 2 for an illustration. At each cut node, we store a counter that indicates the total number of BRDF samples that hit that node.

Compared to the single sampling approach, this new approach incurs a BRDF sampling cost of $O(N_\rho)$, where N_ρ (typically 64) is the number of BRDF samples and is much smaller than the illumination cut size $|\overrightarrow{L_c}|$. As BRDF importance sampling is more effective at using a limited number of samples to resolve high-frequencies, the sampling aliasing problem is greatly reduced. In a way we benefit from decoupling the BRDF sampling from the illumination approximation, making it possible to take advantage of the efficiency in both. On the other hand, our method incurs an additional cost of $O(N_\rho \log(|\overrightarrow{L_c}|))$ for detecting the ray-box intersections. Note that this cost is independent of the cost

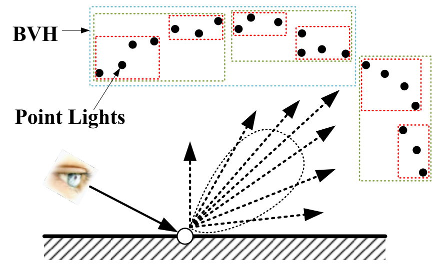


Figure 2: *We detect which cluster(s) a BRDF sample intersects by using a BVH of the light tree.*

for evaluating a BRDF, therefore it's still typically smaller than the single point BRDF sampling cost of $O(|\overrightarrow{L_c}|)$, especially when complex BRDFs are present.

Estimating the per-cluster average BRDF. We now derive the formula to estimate the per-cluster average BRDF based on the BRDF sample count. Assume that among a total of N_ρ BRDF samples, N_ρ^k of them have intersected with cluster $\mathcal{C}_k$. Analytically the average of f_r over $\mathcal{C}_k$ is computed as:

$$\langle \rho_k \rangle = \frac{1}{\Omega_k} \int_{\mathcal{C}_k} f_r(\omega)\, d\omega \tag{6}$$

where Ω_k is the total solid angle subtended by the cluster. For distant illumination, Ω_k is simply $|\mathcal{C}_k| \cdot \frac{4\pi}{|\mathcal{S}|}$. For local illumination, however, the shape of the cluster is not clearly defined, so estimating Ω_k is non-trivial. We use a simple heuristic by treating the cluster as a distant, planar patch perpendicular to the ray direction, thus $\Omega_k = \frac{A_k}{r^2}$, where A_k is the total surface area of the cluster, and r is the distance between the representative light of cluster k to surface point x_o. We clamp this value to an upper limit to avoid significant overestimation as r could be arbitrarily small.

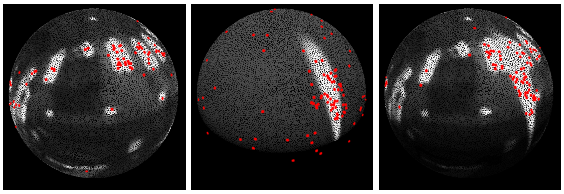
Next, to evaluate $\int_{\mathcal{C}_k} f_r(\omega)\, d\omega$, we note that each BRDF sample, due to importance sampling, contributes equally to the integral (or more intuitively, each sample represents a reflected photon with statistically equal amount of energy). Therefore, the per-sample contribution to the integral is $\frac{1}{N_\rho} \int_{H^2} f_r(\omega)\, d\omega$, meaning each sample shares $\frac{1}{N_\rho}$ of the integral of f_r over the entire hemisphere H^2. Assuming that the BRDF is normalized ($\int_{H^2} f_r = 1$), we have:

$$\int_{\mathcal{C}_k} f_r(\omega)\, d\omega = N_\rho^k \cdot \frac{1}{N_\rho} \int_{H^2} f_r(\omega)\, d\omega = \frac{N_\rho^k}{N_\rho} \tag{7}$$

As predicted, this integral is proportional to the BRDF sample count N_ρ^k stored at each cluster. Now, putting Eq 6, 7, and the estimated Ω_k together, we have:

$$\langle \rho_k \rangle = \frac{r^2 N_\rho^k}{A_k N_\rho} \tag{8}$$

Separating diffuse and specular BRDFs. Our BRDF estimation works more efficiently for highly specular BRDFs than those close to diffuse. This is because diffuse BRDFs



(a) Illumination (b) BRDF (c) Bidirectional

Figure 3: *Visualize distributed samples. Note how bidirectional sampling combines the importance in both* L *and* f_r.

scatter photon energies more widely toward all directions, therefore with a limited number of BRDF importance samples, many clusters may end up getting no samples at all. As a result, the BRDF estimation becomes unreliable. This could be improved by casting more BRDF samples, but at the cost of more expensive computation.

To this end, we use a hybrid approach that separates the treatment of diffuse and specular BRDFs. For specular BRDFs, we use the method described above to estimate $\langle \rho_k \rangle$. For diffuse BRDFs, on the other hand, we simply assign a constant term to each cluster, representing the diffuse energy. This is in fact equivalent to the single sample per-cluster BRDF estimation, which should generally be used when low-frequency BRDFs are present. Since a BRDF contains both specular and diffuse components, we combine the two estimates and derive:

$$\langle \rho_k \rangle = \left(|k_s| \cdot \frac{r^2 N_\rho^k}{A_k N_\rho} + |k_d| \right) \cos(\theta_i^k) \tag{9}$$

where k_s and k_d are the specular and diffuse reflectance parameters. The term $\cos(\theta_i^k)$ accounts for the incident cosine, where θ_i^k is the angle between a cluster's representative light with the normal at a surface point x_o.

3.4. Summary

As a quick summary, to compute the bidirectional sampling distribution, we go through every node on the illumination cut, multiply its value by the BRDF importance estimated using Eq 9, then use the resulting cut as a new PDF to draw visibility samples. All together, the joint probability for drawing a sample is:

$$p(x_s) \sim |\mathcal{C}_k| \langle l_k \rangle \langle \rho_k \rangle \cdot \frac{1}{|\mathcal{C}_k|} = \langle l_k \rangle \langle \rho_k \rangle$$

This effectively accounts for the importance in both illumination and BRDF.

Comparison with SIR. Our method is closely related to the sampling importance resampling (SIR) approach [BGH05]: our illumination clusters are analogous to the initial samples X_i in SIR, which are drawn from one distribution $L(x)$; and the BRDF estimation is analogous to re-weighting X_i to approximate the joint distribution. However, one major difference is that our final samples are not selected from the initial set X_i; instead, they are drawn from the full set of samples by using the joint distribution. Therefore our method is less biased toward the initial distribution L.

To understand the difference, consider the case where SIR draws initial samples from the illumination L, but the BRDF contains a very sharp peak. If the initial sample set is not sufficiently large, the probability that any sample will catch the peak is quite low. As a result, the final samples, selected from the initial set, are unlikely to catch the peak either, resulting in high variance. In our method, due to the use of BRDF importance sampling, the high peaks of BRDF are guaranteed to be caught by some illumination clusters, and hence the importance values of those clusters will be weighted up. Consequently, the final samples will be distributed more within those clusters, resulting in reduced sampling variance. Section 5 provides examples that demonstrate the improvement.

4. Implementation Details

Generating point lights for illumination. Our first step in rendering is to convert illumination to point lights. Details of this step can be found in [AUW07]. For environment lighting, we generate samples on the unit sphere, and for local or indirect lighting, we distribute illumination samples over the scene surface. In both cases we start with random point sampling followed by a repulsion algorithm to distribute points evenly. We follow previous work by choosing $|\mathcal{S}| = 32K$ points, although it's possible for our algorithm to handle a much larger point set.

Computing illumination cut. Our next step is to build a light tree from the point lights by using hierarchical clustering. As long as the geometry of the source remains the same, we do not have to recompute the light tree. The intensities of the point lights can change arbitrarily. To compute an illumination cut, we sample the source radiance L at each frame for all point lights. This is done by either sampling from an environment map, or using a GPU shadow mapper to compute direct lighting at every point. The selection of cut follows the algorithm described in Section 3.2. We typically use a per-cluster L^2 metric to constrain the approximation error, and set $\sigma = 5.0$ in Eq 5. This results in an illumination cut of size between 300 $\sim$ 1000 depending on the actual lighting.

Generating the primary pixel buffer. To reduce the cost of rendering, we use GPU rasterization to generate primary pixels. At each frame we use deferred shading to generate a deep frame buffer which stores the position, normal, and all BRDF parameters that are needed for shading. Using deferred shading makes it easy to apply per-pixel effects such as bump mapping and spatially varying BRDFs.

Computing bidirectional sampling distribution. As described in Section 3.3, the bidirectional sampling distribution is constructed by starting from the illumination cut, and multiplying the luminance value stored at each cut node by an estimate of the average BRDF value. To do so, we need to first send $N_\rho = 64$ sample rays centered at each primary pixel using BRDF importance sampling. We then use the BVH naturally imposed by the light tree to detect the intersection of the sample ray with cut nodes. Note that the ray may intersect with more than one node. Finally, we use the sample count at each node and Eq 9 to estimate the average BRDF.

Final shading. To sample from the bidirectional distribution, we treat the updated cut as a discrete PDF and draw samples from it. Once a cut node is selected, we randomly pick a leaf node from the cluster since all leaf nodes within the cluster represent equal importance. Next, we look up the source radiance at the selected sample, perform a BRDF evaluation, and use ray tracing to compute visibility. The triple product of L, f_r, and V is then accumulated to the image buffer as this sample's contribution to the shading. For simple scenes without much occlusion, we found that as few as $N = 32$ pixel samples is sufficient to produce high-quality images. For complex scenes, however, at least 128 pixel samples is needed to eliminate significant image noise.

5. Results and Discussion

Testing environment. Our tests are performed on a Intel Xeon Quad-Core 2.0 GHz computer with an NVIDIA 8800 GTX graphics card. The steps that generate primary pixel buffers, sample source radiance, build the light tree, and compute shadow mapping are all performed on the GPU. This part of the overhead is negligible compared to the remaining processing, therefore in the following we only report the time it takes to construct the bidirectional sampling distribution and perform the final rendering. Our programs are compiled with Intel Compiler v10.1. We use an unoptimized ray tracer for visibility sampling, and utilize all four cores of the CPU to parallelize the computation. All images are rendered at a default resolution of 640×480.

Test scenes. We have constructed five test scenes: we use the Hebe and Car scenes to test environment lighting, Box and Plate to test local projected lighting, and Bedroom to test indirect lighting (where the direct lighting is a point source). The projected lighting is created by casting a spotlight or HDR projective texture onto an arbitrary portion of the scene. The area under projection indirectly illuminate the rest of the scene. Implementation wise, it is handled in the same manner with indirect lighting, where point lights are distributed over the scene surfaces ahead of time, and are used to sample the direct illumination at run-time. The BRDFs used in each example vary from diffuse to Phong with the exponent parameter up to 100. The Plate scene uses an anisotropic Phong model.

Ground truth images. Our ground truth images are generated by summing up the illumination contribution from all 32K point lights in a brute-force way. We could also use a standard Monte Carlo ray tracer to provide reference images. However, since our method draws samples within the 32K point lights, and converting the illumination to point lights itself introduces some bias, we feel that the ground truth computed our way provides a better base for comparison.

Scene	Faces	L Type	Cut	Samples	Time
Hebe	64 K	Env.	420	32	10 s
Car	30 K	Env.	648	128	28 s
Box	20 K	Proj.	267	128	27 s
Plate	1 K	Proj.	492	128	29 s
Bed	95 K	Ind.	986	256	69 s

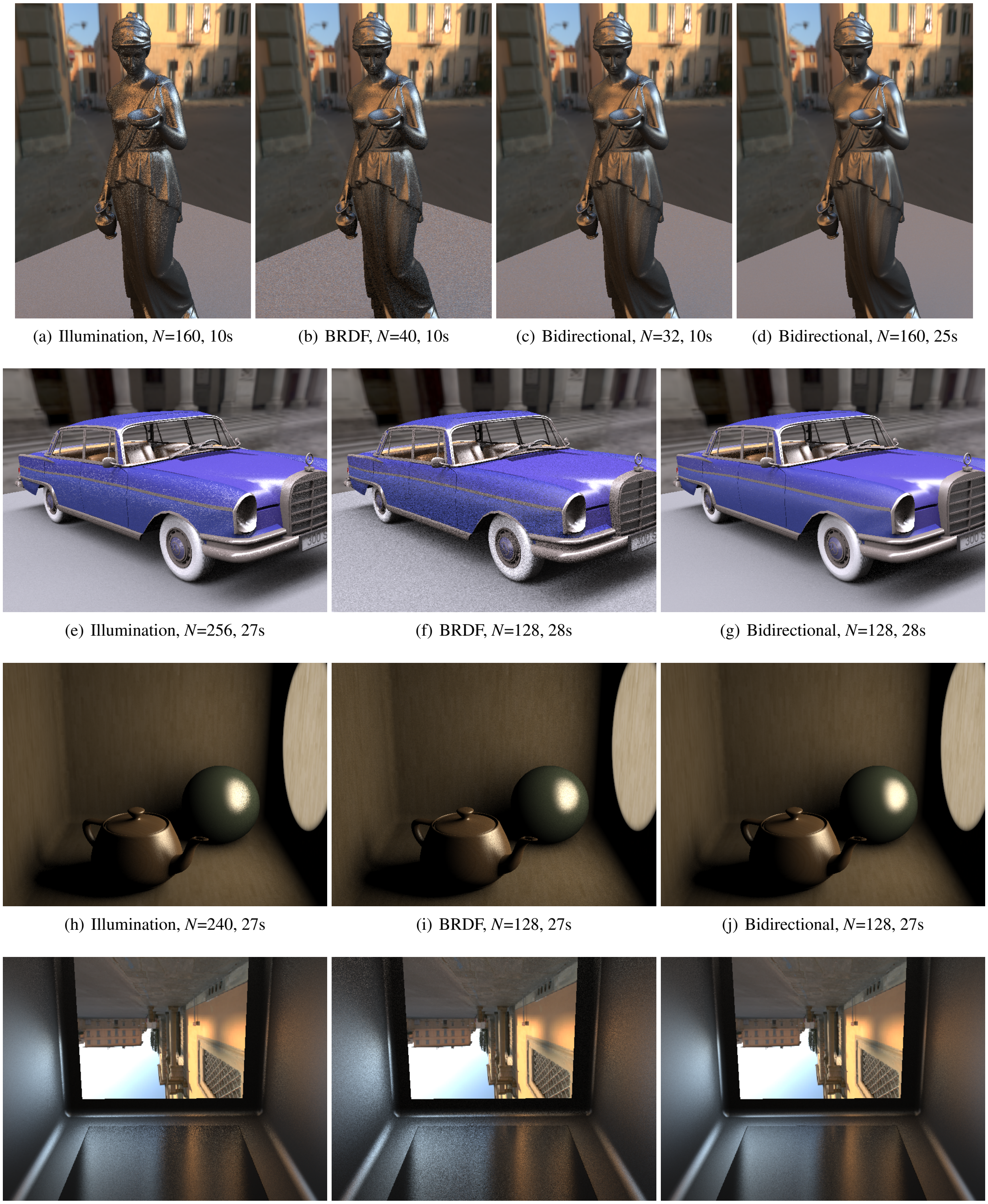
Figure 4: *Test profiles. The columns list the number of faces of each scene, illumination type, illumination cut size, pixels samples used for final rendering, and rendering time.*

Quality and performance. In figure 5 we show comparisons of rendering quality by using illumination importance sampling, BRDF importance sampling, and our bidirectional importance sampling respectively. We carefully set the number of pixel samples in each case such that the total computation time for each is approximately the same. All images are computed at 640×480 resolution (except the Hebe scene is at 480×640 resolution). Note that the illumination importance sampling is often $2 \sim 3$ times faster than the other two choices, because no BRDF evaluation is required. However, in the presence of glossy BRDFs, purely illumination based sampling converges very slowly. Even with a large number of pixel samples, it still has difficulties matching the quality of bidirectional sampling. Pure BRDF importance sampling, on the other hand, face efficiency problems when the scene contains smooth BRDFs but high-frequency lighting. This can be seen from several examples we provided, particularly in the diffuse ground floor at each example.

In Figure 7 we show the bedroom scene with one bounce of indirect lighting. The direct lighting comes from a single point light and is computed on the GPU. The global illumination of this scene is dominated by indirect lighting. Again, the first three columns are computed in approximately the same time. From the insets of the second row (indirect lighting), we can clearly see the advantage of using bidirectional importance sampling. A summary of all scenes and the performance data can be found in Figure 4.

Sample distribution. To understand how the importance function affects the distribution of samples, we visualize the sampling result in Figure 3. This result is generated for a single pixel placed at the center of a plane under environment lighting. An anisotropic Phong BRDF is used for BRDF sampling. As seen from the figure, bidirectional importance sampling draws samples according to the joint distribution of the illumination and BRDF, while the other two examples draw samples only from a single distribution.

Comparison with SIR. Figure 8 shows a comparison of our method with SIR [BGH05, TCE05]. We implement SIR as follows: we first draw $M = 800$ initial samples X_i from all 32K point lights according to illumination importance;



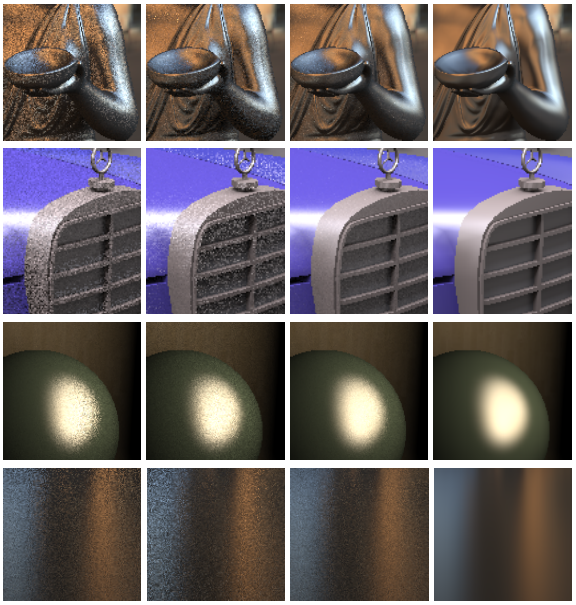
(a) Illumination, N=160, 10s (b) BRDF, N=40, 10s (c) Bidirectional, N=32, 10s (d) Bidirectional, N=160, 25s

(e) Illumination, N=256, 27s (f) BRDF, N=128, 28s (g) Bidirectional, N=128, 28s

(h) Illumination, N=240, 27s (i) BRDF, N=128, 27s (j) Bidirectional, N=128, 27s

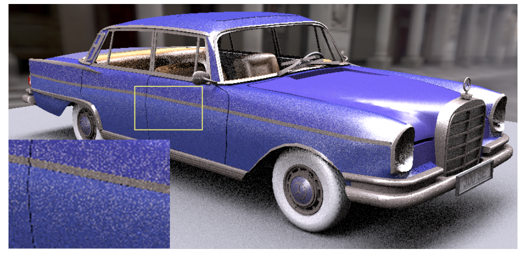
(k) Illumination, N=260, 28s (l) BRDF, N=128, 27s (m) Bidirectional, N=128, 29s

Figure 5: *Comparison of illumination importance sampling, BRDF importance sampling, and our bidirectional importance sampling. Each set of images is computed at approximately the same time. N is the number of pixel samples used in final rendering. Note that illumination importance sampling performs poorly when material is highly glossy, and BRDF importance sampling performs poorly when material is diffuse. In contrast, bidirectional sampling gives better results in both cases.*

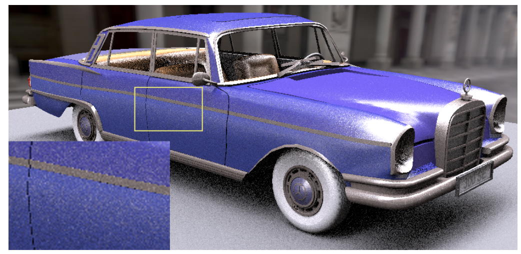


Illumination BRDF Bidirectional Reference

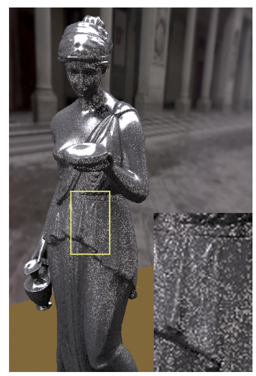
Figure 6: *Blow-up images from Figure 5. The first three columns in each set are computed at the same amount of time, and the last column shows reference images computed using a brute-force approach.*



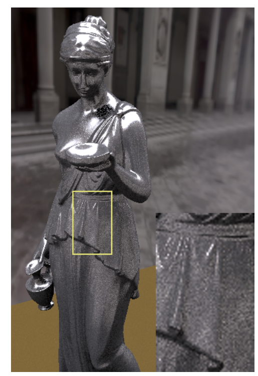
(a) SIR, $M = 800$, $N = 32$, 960×480 image, 30s



(b) Ours, $|L_c| = 800$, $N = 32$, 960×480 image, 26s



(c) SIR, 400×600 res., 8s



(d) Ours, 400×600 res., 7s

Figure 8: *Comparison of SIR with our method.*

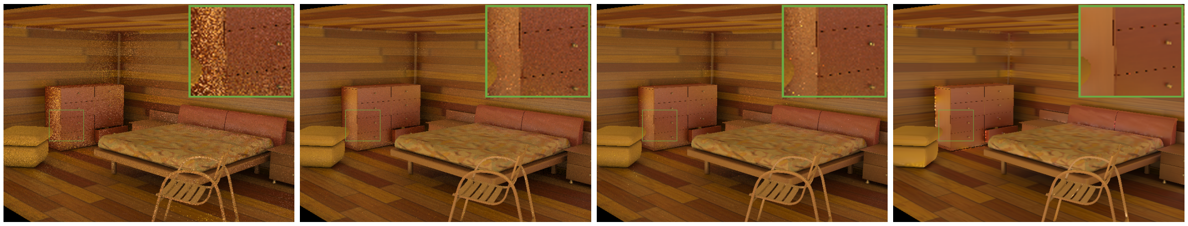
we then evaluate the BRDF (with the incident cosine term) at these samples, and draw $N = 32$ final samples from them using the evaluated BRDF values. To compare, we use our method to generate an illumination cut of equal size 800, then draws 32 final samples after the cut is updated with per-cluster BRDF average.

In the first example (Car), the illumination comes primarily from the top, causing SIR to distribute initial samples X_i mainly in that area. Because SIR selects final samples from X_i, it leads to significant noise at pixels on the side of the car, as most samples in X_i do not contribute to those pixels. With our method, while the illumination cut does devote more clusters to the top area, the clusters on the side receive more BRDF importance samples in this case. Consequently, the final samples are distributed more efficiently towards those clusters and the sampling variance is reduced. The second example (Hebe) includes a very glossy BRDF (Phong with exponent 100). This causes similar problems for SIR, as the initial set X_i is blind to the high peaks of the BRDF. Note that in both examples, due to the large number of BRDF evaluations (800/pixel), SIR performs slightly worse than our method.

6. Conclusion and Future Work

In summary, this paper presents a new method for bidirectional importance sampling of unstructured illumination. We combine an illumination cut with BRDF importance sampling to efficiently compute a bidirectional importance sampling function. As a main advance over previous work, our method allows for unstructured light sources. In our current system, we found that sampling and evaluating BRDF on the CPU take a great amount of time. Therefore in the future we would like to exploit the GPU to accelerate this computation. Eventually by having an efficient GPU ray tracer, we can move the entire pipeline to the GPU. In addition, we would like to explore the use of our approach in other illumination effects such as subsurface scattering. Finally, the ideas from this paper may be combined with other work, such as the Lightcuts, to provide a more efficient BRDF estimate.

Acknowledgements The authors would like to thank Paul Debevec for his light probe images. This work was supported in part by NSF grant CCF-0746577 and the UMass FRG grant P1-FRG-0029. Oskar Åkerlund was sponsored by Professor Anders Ynnerman at Linköping University.



(a) Illumination, *N*=320, 70s (b) BRDF, *N*=256, 68s (c) Bidirectional, *N*=256, 69s (d) Reference

Figure 7: *The bedroom scene. Only the indirect illumination component is shown. The direct lighting comes from a point light. The change in image noise can be clearly seen from the insets.*

References

[ARBJ03] Agarwal S., Ramamoorthi R., Belongie S., Jensen H. W.: Structured importance sampling of environment maps. *ACM Trans. Graph. 22*, 3 (2003), 605–612.

[AUW07] Akerlund O., Unger M., Wang R.: Precomputed visibility cuts for interactive relighting with dynamic BRDFs. In *Proceedings of Pacific Graphics* (2007), pp. 161–170.

[AWB08] Arbree A., Walter B., Bala K.: Single-pass scalable subsurface rendering with lightcuts. *Computer Graphics Forum 27*, 2 (2008), 507–516.

[BGH05] Burke D., Ghosh A., Heidrich W.: Bidirectional importance sampling for direct illumination. In *Proceedings of Eurographics Symposium on Rendering* (2005), pp. 147–156.

[CAM08a] Clarberg P., Akenine-Möller T.: Exploiting Visibility Correlation in Direct Illumination. *Computer Graphics Forum 27*, 4 (2008), 1125–1136.

[CAM08b] Clarberg P., Akenine-Möller T.: Practical product importance sampling for direct illumination. *Computer Graphics Forum 27*, 2 (2008), 681–690.

[CD] Cohen J., Debevec P.: The lightgen plugin, http://www.hdrshop.com/main-pages/plugins.html.

[CETC06] Cline D., Egbert P. K., Talbot J. F., Cardon D. L.: Two stage importance sampling for direct lighting. In *Proceedings of Eurographics Symposium on Rendering* (2006), pp. 103–113.

[CJAMJ05] Clarberg P., Jarosz W., Akenine-Möller T., Jensen H. W.: Wavelet importance sampling: efficiently evaluating products of complex functions. *ACM Trans. Graph. 24*, 3 (2005), 1166–1175.

[CPWAP08] Cheslack-Postava E., Wang R., Akerlund O., Pellacini F.: Fast, realistic lighting and material design using nonlinear cut approximation. *ACM Trans. Graph.* (2008), to appear.

[Deb05] Debevec P.: A median cut algorithm for light probe sampling. In *ACM SIGGRAPH 2005 Posters* (2005), p. 66.

[DM97] Debevec P. E., Malik J.: Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH '97* (1997), pp. 369–378.

[DWF06] Donikian M., Walter B., Fernandez S.: Accurate direct illumination using iterative adaptive sampling. *IEEE Trans. Visualization and Computer Graphics 12*, 3 (2006), 353–364.

[GH06] Ghosh A., Heidrich W.: Correlated visibility sampling for direct illumination. *Visual Computer 22*, 9 (2006), 693–701.

[HP02] Hey H., Purgathofer W.: Importance sampling with hemispherical particle footprints. In *Proc. of the 18th spring conference on Computer Graphics* (2002), pp. 107–114.

[HPB07] Hašan M., Pellacini F., Bala K.: Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. 26*, 3 (2007), 26.

[Jen95] Jensen H. W.: Importance driven path tracing using the photon map. In *Proceedings of Eurographics Rendering Workshop* (1995), pp. 326–335.

[Kaj86] Kajiya J. T.: The rendering equation. In *Proceedings of SIGGRAPH '86* (1986), pp. 143–150.

[KK03] Kollig T., Keller A.: Efficient illumination by high dynamic range images. In *Proceedings of Eurographics Symposium on Rendering* (2003), pp. 45–50.

[LRR04] Lawrence J., Rusinkiewicz S., Ramamoorthi R.: Efficient BRDF importance sampling using a factored representation. *ACM Trans. Graph. 23*, 3 (2004), 496–505.

[NRH04] Ng R., Ramamoorthi R., Hanrahan P.: Triple product wavelet integrals for all-frequency relighting. *ACM Trans. Graph. 23*, 3 (2004), 477–487.

[ODJ04] Ostromoukhov V., Donohue C., Jodoin P.-M.: Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph. 23*, 3 (2004), 488–495.

[PH04] Pharr M., Humphreys G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[Pha] Pharr M.: Extended photon map implementation, http://www.pbrt.org/plugins/exphotonmap.pdf.

[SL06] Steinhurst J., Lastra A.: Global importance sampling of glossy surfaces using the photon map. In *IEEE Symposium on Interactive Ray Tracing* (2006), pp. 133–138.

[TCE05] Talbot J., Cline D., Egbert P.: Importance resampling for global illumination. In *Proceedings of Eurographics Symposium on Rendering* (2005), pp. 139–146.

[Vea98] Veach E.: *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford, CA, USA, 1998.

[WABG06] Walter B., Arbree A., Bala K., Greenberg D. P.: Multidimensional lightcuts. *ACM Trans. Graph. 25*, 3 (2006), 1081–1088.

[WFA*05] Walter B., Fernandez S., Arbree A., Bala K., Donikian M., Greenberg D. P.: Lightcuts: a scalable approach to illumination. *ACM Trans. Graph. 24*, 3 (2005), 1098–1107.