Eurographics Symposium on Rendering (2005)
Kavita Bala, Philip Dutré (Editors)

Metropolis Photon Sampling with Optional User Guidance

Shaohua Fan†, Stephen Chenney and Yu-Chi Lai

University of Wisconsin, Madison

Abstract

We present Metropolis Photon Sampling *(MPS), a visual importance-driven algorithm for populating photon maps. Photon Mapping and other particle tracing algorithms fail if the photons are poorly distributed. Our approach samples light transport paths that join a light to the eye, which accounts for the viewer in the sampling process and provides information to improve photon storage. Paths are sampled with a Metropolis-Hastings algorithm that exploits coherence among important light paths. We also present a technique for including user selected paths in the sampling process without introducing bias. This allows a user to provide hints about important paths or reduce variance in specific parts of the image. We demonstrate MPS with a range of scenes and show quantitative improvements in error over standard Photon Mapping and Metropolis Light Transport.*

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

Keywords: *global illumination, Photon Mapping, particle tracing, visual importance, MCMC, user input*

1. Introduction

Applications from film special effects to industrial design demand realistic renderings of complex scenes, yet the accurate computation of global illumination remains a challenging problem. Photon Mapping [Jen01] is the current choice of industry for scenes with general surface primitives and reflectance functions [Dri01]. It uses an initial pass to populate photon maps with samples of the power arriving at points in the scene. A *final gather* pass then uses the maps to estimate the contribution of indirect illumination to visible pixels. It is essential that the initial pass populate the maps with photons useful to the final gather, but the standard technique fails to do so in some common scenes. This paper presents *Metropolis Photon Sampling* (MPS), a Monte Carlo sampling algorithm for constructing photon maps that produces high-quality results in situations where standard photon map construction fails. MPS also gives users a technique to control variance over the image.

Standard Photon Mapping traces particles from the lights distributed according to the lights' power distribution, and deposits photons when the particles interact with surfaces. It performs poorly when little of the lights' total power arrives at locations important to the final gather. This situation is not uncommon in practice: indoor environments may have many lights that contribute unevenly to the image (Figure 1); in some scenes most light paths are occluded (Figure 4); and local views of outdoor scenes may see little of the sun's power (under a forest canopy or in downtown city streets). Poor sampling results in excess noise in the indirect illumination estimates derived from the map. Furthermore, low photon density leads to larger search radii in accessing photons, which causes inappropriate samples to be included and hence severe energy bleeding. Both effects are evidenced in the left image of Figure 1, based on the photon distribution on the left in Figure 2.

One underlying cause of a poor sample distribution is the lack of visual importance information; sampling from the light does not consider the camera location. Our first contribution is a technique, *Metropolis Photon Sampling* (MPS), that builds photon maps using complete light paths that join a light to the eye. By linking to the eye we account for visual importance and can identify photon storage locations that will be useful to the final gather (Figure 2). This re-

† shaohua@cs.wisc.edu

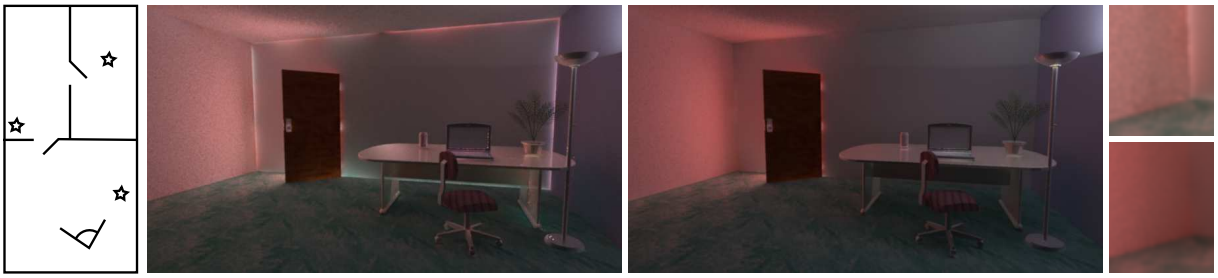

Figure 1: *Leftmost is the plan of a scene in which only a small portion of the lights' total power contributes to the image. The left image was produced using standard Photon Mapping, which under-samples some regions and over-samples others, resulting in image noise and severe energy bleeding from the adjacent room (the cause of the incorrect illumination around the edges of the rear wall). To the right is our result. Paths joining the eye to a light were sampled and photons were stored only in important locations. The insets on the far right show zoomed sections taken from the center-left of the images, and demonstrate how our method (lower) both reduces noise and avoids energy bleeding.*

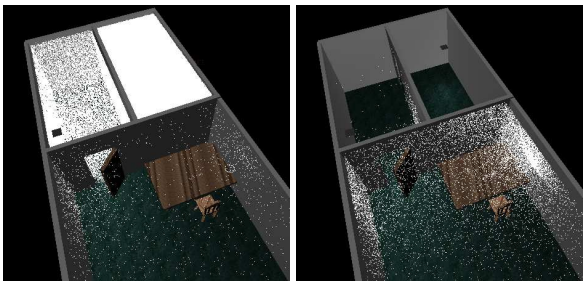

Figure 2: *Photon distributions for Figure 1. While standard Photon Mapping generates many photons in a short period of time (left), they are almost all located in places not relevant to the final image. Right is our result for identical computation time, with all the samples in locations useful to a final gather operation.*

duces image noise and energy bleeding artifacts in scenes where most paths traced only from the lights are irrelevant to the image (Figure 1). MPS uses a Metropolis-Hastings algorithm [MRR*53, Has70, GRS96] to sample over paths, but the general framework supports other sampling methods.

Regardless of the sampling strategy used, light paths that are difficult to find randomly lead to image artifacts in Monte Carlo rendered images. In Photon Mapping this tends to manifest itself as smooth but incorrect results, while in a pure Monte Carlo framework the result is noise. Frequently the difficult paths are obvious to a user: light may have to pass through a small opening or be focused by a particular scene element. Our second contribution enables a user to provide a small set of important light transport paths that the sampling process uses to reduce variance. No bias is introduced to the result. User defined paths help when sampling from difficult geometric arrangements, and also give a user local control over variance in the image. For instance, in Figure 1 the user suggested 10 paths that carry light through the doorway from the neighboring room. Ours is the first technique in the rendering literature for including specific user-defined sample paths in a Monte Carlo framework.

2. Related Work

The rendering equation [Kaj86, PH04] is the physical foundation for image synthesis. Here we concentrate on Monte Carlo methods for solving the equation; while finite element methods for general scenes have been proposed [ICG86, SP89, WCG87], they are not widely used due to high computation time and memory requirements. The first unbiased Monte Carlo *path tracing* algorithm was introduced by Kajiya [Kaj86]. Path tracing builds random ray trees rooted at the eye and considers each valid transport path as a sample. Bi-directional path tracing [LW93, VG94] forms paths from both the eye and a light and joins them. This has the advantage of combining both visual importance and the lights' power, but the disadvantage that each path is independent; while a difficult path may be located by random chance, it cannot be further exploited. Kollig and Keller [KK00] address this problem with quasi-Monte Carlo methods, which can exploit coherence in random number space under the assumption that paths generated with similar random choices are similar paths, which is not necessarily the case in even mildly complex scenes.

Veach [VG97] presented *Metropolis Light Transport* (MLT), which is a Markov chain Monte Carlo (MCMC) algorithm designed to exploit coherence in path space. MCMC views sampling as a Markov process, and hence a good sample found in one step can improve subsequent samples. MCMC also allows multiple sampling strategies to be combined without introducing bias, which enables us to incorporate user-guided sampling. Veach's *Multiple Importance Sampling* [VG95] also combines different strategies, and it could also support user input of the form we propose. An alternate MCMC approach to rendering has been proposed by Kelemen et al. [KSKAC02]. Rather than sampling in path space, they sample on a high-dimensional unit cube. MLT

was extended to support participating media by Pauly et al. [PKK00].

A single sample may be representative of illumination over a large region if radiance varies slowly, as is often the case in scenes with significant indirect diffuse illumination. *Particle tracing* algorithms, of which Photon Mapping is one, exploit this to re-use light paths. Arvo [Arv86], Heckbert [Hec90] and Collins [Col94] proposed algorithms that use *illumination-maps* to store irradiance arriving along sampled paths. Like Photon Mapping, particles are traced from the lights, but they require parameterized geometry for the maps. The method of Shirley et al. [SWH*95] traces particles and builds a polygonal mesh representation that can be rendered in real time for varying viewpoint. Chen et al. [CRMT91] also worked with 2D maps but in addition offered a progressive refinement solution. Our sampling method could be used with any of these prior techniques, with some modification to particle storage. Ward's *RADIANCE* system [WRC88, War94] traces rays from the eye and caches diffuse contributions for use in subsequent estimates. The *irradiance caching* technique [WH92] is used to determine if the cached samples provide an adequate estimate.

Many rendering algorithms have been developed to exploit visual importance; see Christensen [Chr03] for a survey. Specific to particle tracing, *importon* techniques trace particles from the eye to construct an *importon map* that is used to estimate visual importance. Peter and Pietrek [PP98] used the importon map to construct importance sampling distributions for each scattering event of the particle tracing phase. The algorithm is expensive due to the cost of computing distributions at every particle bounce, its local decisions may not produce a globally important path, and the importance sampling produces photons with highly variable power. Keller and Wald [KW00] used importon maps to avoid photon storage in areas that contribute little to the final image. Their technique reduces memory usage and maintains roughly uniform photon power, but gives no control over the generation of the samples in the first place. Suykens and Willems' [SW00] algorithm considers the current sample density in the photon map when storing a new sample and redistributes its power if it would result in excess density (without modifying photon generation). Unlike existing methods, our algorithm samples from complete paths joining the light to the eye and thus efficiently accounts for visual importance without using importons. Complete paths also provide information about important photon storage locations and hence reduce redundant photons.

Variance is typically controlled by using more samples, or designing new algorithms (not a natural tool for most end-users). Ward [War94] allows users to specify surfaces as important secondary light sources, and the system builds their outgoing irradiance functions for use in indirect illumination. The technique is targeted at large secondary sources, such as windows, but fails if the secondary source itself is not easy to reach from the light or no one surface is significant enough to warrant the attention. Our approach allows a user to specify paths through multiple reflections, and places no restrictions on the surfaces or pieces of surface affected. A related idea to user input is sampling based on pilot paths that are found in a random initial pass (or in the previous frame of an animation). Dmitriev et al. [DBMS02] discuss this approach in the animation context, but it relies on similarity in random number space to compute path perturbations. With user input, there are no random variables associated with the paths, so this approach cannot be applied.

3. Light Paths To Photons

We incorporate visual importance into photon map construction by extracting photons from *complete* light paths that join a point on a light source to the eye through some number of scattering (reflection or transmission) events. Complete paths also allow us to identify the point on the path at which a photon should be stored. Assume for the moment that we can produce sample light paths. In this section we address the way in which photons are extracted from the paths.

3.1. Photon Locations

Given a light path, we wish to identify the point or points along it that will be accessed during a photon map lookup. This clearly depends on how the final gather is performed. We use a standard Photon Mapping final gather as described by Jensen [Jen01], to whom we refer the reader for motivation and details. Estimation of radiance from the global photon map takes place at points that lie at the second diffuse bounce on paths traced from the eye (possibly with intervening specular bounces). Hence, we store a photon at the second diffuse point for each path that our sampler produces. Estimation from caustic photons occurs at the first diffuse bounce, so we store in both the global and caustic map at the first diffuse point along caustic paths. In any case, we refer to the photon storage location on a path as the *storage point*.

The nearest neighbors around a point of interest, $\mathbf{p}$, are used when estimating radiance from the maps. The neighbors are assumed to be representative of the incoming radiance at $\mathbf{p}$, which requires that radiance vary slowly in the region from which they come. This assumption is more likely to be true, and hence the estimate better, as the density of photons around $\mathbf{p}$ increases and the neighbors fall within a smaller region. Our algorithm ensures that most stored photons lie around points where final gather estimates are formed, and hence optimizes the quality of the estimate for a given map-building effort and memory footprint.

The use of a *kd*-tree for photon storage removes the need for a surface parameterization (allowing for a wider range of surfaces and fast neighbor lookup) but this also decouples photons from surface properties. Severe light bleeding can occur due to the breakdown of the slow varying radiance

assumption, which is hard to detect without surface information. This is a major problem in scenes where a light is on the back side of a thin divider, as in Figure 1.

A common practical solution is to store a normal vector with each photon and require that it be similar to the normal at the point where the estimate is being taken. This reduces bleeding in concave corners, but fails in our scenes. For instance, the floor is oriented the same on both sides of the wall in Figure 1. Importon based methods (Section 2) fail to address the energy bleeding through walls problem because importance can leak just as energy does, allowing photons to be stored in unimportant regions. However, points on the back side of a wall are almost never the second diffuse bounce on a path from the eye, so our method automatically avoids storing them and hence significantly reduces energy bleeding in from unimportant areas of the scene.

3.2. Photon Storage

As with standard Photon Mapping, for each photon j, we store the location, $\mathbf{x}^{(j)}$, incoming ray direction, $\theta^{(j)}$, and radiant flux (power), $\Phi^{(j)}$. In this section we describe how $\Phi^{(j)}$ is computed for a sampled path. Our discussion is based on the particle tracing framework introduced by Veach [Vea97, §4.A] and applied to Photon Mapping by Pharr and Humphreys [PH04]. Expressed in terms of sampling from path space, we require that each photon j have power such that

$$E\left[\sum_{\mathcal{R}} \Phi^{(j)}\right] = \int_{\Omega_{\mathcal{R}}} f_{map}(\overline{\mathbf{x}}) \mathrm{d}\mu(\overline{\mathbf{x}}) \qquad (1)$$

where the sum is over the set of photons within a region of area and solid angle, $\mathcal{R}$. The integral is over $\Omega_{\mathcal{R}}$, the set of light transport paths that begin on a light and end within the region, $\mu(\overline{\mathbf{x}})$ is the surface area measure for the path $\overline{\mathbf{x}}$, and $f_{map}(\overline{\mathbf{x}})$ is defined as

$$f_{map}(\overline{\mathbf{x}}) = L_e(\mathbf{x}_0,\mathbf{x}_1)G(\mathbf{x}_0,\mathbf{x}_1) \cdot \prod_{i=1}^{m-1} f_s(\mathbf{x}_{i-1},\mathbf{x}_i,\mathbf{x}_{i+1})G(\mathbf{x}_i,\mathbf{x}_{i+1})$$

in which $\mathbf{x}_i$ is a point on the path $\overline{\mathbf{x}}$, $L_e(\mathbf{x}_0,\mathbf{x}_1)$ is the radiance emitted by a light point $\mathbf{x}_0$ toward $\mathbf{x}_1$, $f_s(\mathbf{x}_{i-1},\mathbf{x}_i,\mathbf{x}_{i+1})$ is the bidirectional scattering distribution function for surface point $\mathbf{x}_i$, and $G(\mathbf{x}_i,\mathbf{x}_{i+1})$ is the geometry term between points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$:

$$G(\mathbf{x}_i,\mathbf{x}_{i+1}) = V(\mathbf{x}_i,\mathbf{x}_{i+1}) \frac{|\cos(\theta_i)\cos(\theta_i')|}{\|\mathbf{x}_i - \mathbf{x}_{i+1}\|^2}$$

θ_i and θ_i' are the angles between $\mathbf{x}_i \rightarrow \mathbf{x}_{i+1}$ and the surface normals at $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ respectively. The visibility term $V(\mathbf{x}_i,\mathbf{x}_{i+1})$ has value 1 if $\mathbf{x}_i$ can see $\mathbf{x}_{i+1}$ and 0 otherwise.

If we consider the region of interest, $\mathcal{R}$, to be all the points accessed during the final gather, Equation 1 takes the form of a Monte Carlo estimate of an integral. The sum on the left is over all the photons in the map, and the integral on the right evaluates to the total power arriving in the map, B_{map}. If we sample paths according to the distribution $p_{map} = f_{map}(\overline{\mathbf{x}})/B_{map}$, each one of the N photons should have the same power: $\Phi = B_{map}/N$.

We only store photons at points relevant to the final gather, so the above discussion assumes we are sampling over paths terminating at such points. However, the designation of storage points relies on having the complete path to the eye, in order to count the number of diffuse bounces on the subpath from the eye. To obtain this information, MPS samples from the space of all paths that join the light to the eye and stores photons only for the desired sub-paths. We sample according to the probability distribution function (pdf) given by $p_{eye}(\overline{\mathbf{x}}) = f_{eye}(\overline{\mathbf{x}})/B_{eye}$, where

$$f_{eye}(\overline{\mathbf{x}}) = W(\overline{\mathbf{x}})L_e(\mathbf{x}_0,\mathbf{x}_1)G(\mathbf{x}_0,\mathbf{x}_1) \cdot \prod_{i=1}^{m-1} f_s(\mathbf{x}_{i-1},\mathbf{x}_i,\mathbf{x}_{i+1})G(\mathbf{x}_i,\mathbf{x}_{i+1}) \qquad (2)$$

The function $W(\overline{\mathbf{x}})$ takes the value 1 if the path passes through the image plane, and 0 otherwise. B_{eye} is the normalizing constant, in this case the total power arriving at the image, and should satisfy

$$B_{eye} = \int_{\Omega_{eye}} W(\overline{\mathbf{x}}) f_{eye}(\overline{\mathbf{x}}) d\mu(\overline{\mathbf{x}})$$

where Ω_{eye} is the space of all paths that join a light to the eye. Following Veach [Vea97], we use path tracing to estimate this integral. Not many path tracing samples are required because we are averaging across all pixels.

When we use p_{eye} as the target distribution the resulting samples will no longer be distributed according to p_{map} as required for correct photon map estimation (Equation 1). This is accounted for using standard importance sampling re-weighting:

$$\Phi^{(j)} = \frac{1}{N}\frac{f_{map}(\overline{\mathbf{x}}^{(j)}_{map})}{p_{eye}(\overline{\mathbf{x}}^{(j)})} = \frac{B_{eye}}{N}\frac{f_{map}(\overline{\mathbf{x}}^{(j)}_{map})}{f_{eye}(\overline{\mathbf{x}}^{(j)})}$$

where $\overline{\mathbf{x}}_{map}$ is the sub-path $L(D|S)^*D$ from a sampled path of the form $L(D|S)^*DS^*DS^*E$ for which a photon is stored in the global map, or the sub-path LS^*D of an LS^*DS^*E path for caustic photon storage. Note that we no longer require B_{map}. Furthermore, when sampling according to $p_{eye}(\overline{\mathbf{x}})$ we may generate paths that do not result in photon storage (i.e. not of the form $L(D|S)^*DS^*DS^*E$ or LS^*DS^*E). In this case, $f_{map} = 0$ and no photon is stored.

The Metropolis-Hastings sampler we use may provide many paths with the same storage point, $\mathbf{x}^{(j)}$, and incoming ray direction, $\theta^{(j)}$. This is due either to rejection of candidate paths, in which case the entire path is repeated, or a proposal that retains the storage point while changing some other part of the path (see Section 4). Instead of generating a

new photon in such cases, we accumulate the power in a single photon and hence reduce photon storage cost and look-up time. In practice, few paths contribute to any one photon and the resulting per-photon power variation does not create artifacts.

The scattering function $f_s(\mathbf{x}_{i-1}, \mathbf{x}_i, \mathbf{x}_{i+1})$ is wavelength dependent. We evaluate f_s for the standard RGB channels, and use them to compute $f_{map,R}$, $f_{eye,R}$, etc. For the sampling process we must attach a single probability to each path. We use the luminance channel, $f_{eye,Y}$, computed with the RGB to XYZ color conversion. With this path probability, the red power for the stored photon (green and blue are similar) is

$$\Phi_R^{(j)} = \frac{B_{eye,Y}}{N} \frac{f_{map,R}(\overline{\mathbf{x}}_{map}^{(j)})}{f_{eye,Y}(\overline{\mathbf{x}}^{(j)})}$$

The framework developed to this point does not depend on the method for finding sample paths, or even on their pdf, p_{eye}. Any sampling technique capable of generating paths from the light to the eye, such as bi-directional path tracing, could be used. We chose a Metropolis-Hastings sampler because it can both exploit coherence in path space and support user input.

4. Sampling Paths

Metropolis-Hastings algorithms use a Markov process designed to obtain a sequence of samples whose distribution converges to a target pdf. Following Veach [Vea97], to estimate radiometric quantities we want each sample path, $\overline{\mathbf{x}}$, to come from the space of all transport paths joining the light to the eye, Ω_{eye}. The target pdf is $p_{eye}(\overline{\mathbf{x}})$. Each path $\overline{\mathbf{x}}$ with m segments is parameterized by the surface intersection points at which a scattering event occurs, $\mathbf{x}_i$, $i \in [1, \ldots, m-1]$, along with the final point, $\mathbf{x}_m$, and the point on the light source from which the particle is emitted, $\mathbf{x}_0$.

The Markov process generates each sample in the sequence, X_t, by proposing a candidate, X_t', based on the previous sample X_{t-1}, and either accepting this candidate as X_t or rejecting it and repeating X_{t-1}. In pseudo-code:

$X_0 \leftarrow$ *initialSample*()
for $t = 1$ to N
  $X_t' \leftarrow$ *propose*(X_{t-1})
  $r \leftarrow$ *uniformRandom*$[0,1)$
  if ($r < \alpha(X_t'|X_{t-1})$) then
    $X_t = X_t'$
  else
    $X_t = X_{t-1}$

The procedure *initialSample* chooses one of the paths generated by the path tracing computation for B_{eye}, according to the distribution p_{eye}. The initial sample chosen in this way is unbiased, so there will be no start-up bias in the Markov chain [GRS96, Vea97]. The proposal function, *propose*(X_{t-1}), produces a new light path by applying a random modification to the current sample. While the correctness conditions placed on the modifications are not difficult to satisfy, the strategies employed are the primary factor in determining the efficiency of the algorithm (the number of samples required for a good estimate). We describe our mutation strategies below.

The function $\alpha(X_t'|X_{t-1})$ computes the *acceptance probability* for X_t' given the current sample.

$$\alpha(X_t'|X_{t-1}) = \min\left\{1, \frac{f_{eye,Y}(X_t')T(X_{t-1}|X_t')}{f_{eye,Y}(X_{t-1})T(X_t'|X_{t-1})}\right\} \quad (3)$$

The function $f_{eye,Y}(X_t')$ is proportional to the target pdf $p_{eye}(\overline{\mathbf{x}})$ (and the normalization constant cancels out).

$T(X_t'|X_{t-1})$ is the transition function (or proposal distribution) which gives the probability of choosing, by any means, X_t' given X_{t-1}. Note that the reverse transition function, $T(X_{t-1}|X_t')$, is also required, and in a Metropolis-Hastings sampler it need not equal $T(X_t'|X_{t-1})$.

4.1. Proposal Strategies

The techniques used in the *propose*(X_{t-1}) procedure of the MCMC algorithm are the key to its efficient and correct operation. There are two conflicting goals in designing a good proposal. The candidate path, X_t', should be as different as possible from the current path, X_{t-1}, to rapidly move around the sample state space. At the same time it should be sufficiently similar to X_{t-1} to exploit coherence in high-power paths. The technical conditions on *propose*(X_{t-1}) ensure that there is some non-zero probability way to move between any two non-zero probability paths (see Gilks et al. [GRS96]). The acceptance probability, $\alpha(X_t'|X_{t-1})$, is specifically designed to take *any* proposal strategy that meets the conditions, properly encoded in the transition functions $T(X_{t-1}|X_t')$ and $T(X_t'|X_{t-1})$, and create an unbiased sampler.

We introduce two novel mutation strategies. **User Path (UP)** proposals make use of user hints about which paths are likely to be important to the final result (Section 5). The variance of any estimate is reduced around the given paths. **Photon Map (PM)** proposals explore paths that will contribute to the global photon map (Section 5.3). They change the sample path while retaining the DS^*E sub-path to the eye.

In addition, four other proposal types previously described for MLT are suitable for use here [VG97]. **Bi-Directional (BD)** proposals modify sub-paths of the current path, with the aim of rapidly exploring the sampling space. **Caustic Perturbation (CP)** and **Lens Perturbation (LP)** proposals also modify sub-paths, but this time with the aim of exploiting coherence in high-power, localized features. Finally, **Lens Sub-path (LS)** proposals stratify samples across the image, which ensures that enough samples

are captured in darker regions of the scene. We implement each of these strategies in the same manner as MLT.

Each time the $propose(X_{t-1})$ procedure is called we choose one of the above strategies at random according to a fixed distribution. That is, $propose_{type}(X_{t-1})$ is selected with probability P_{type} where $type$ is one of the above options and $\sum_{type} P_{type} = 1$. In computing the transition function, $T(X'_t|X_{t-1})$, all possible proposals that might generate X'_t from X_{t-1} should be considered:

$$T(X'_t|X_{t-1}) = \sum_{type} P_{type} T_{type}(X'_t|X_{t-1}) \quad (4)$$

However, it is also acceptable to consider only the function derived from the proposal strategy chosen to generate X'_t [Tie98, AD99]:

$$T(X'_t|X_{t-1}) = T_{chosen}(X'_t|X_{t-1}) \quad (5)$$

We use a combination of both strategies: Equation 5 avoids the computation of unnecessary transition functions, but Equation 4 is required for user path proposals (Section 5.2).

5. User Path Proposals

The user path proposal strategy increases the proportion of candidate paths around those supplied by the user. This results in variance reduction for any estimate based on the paths, such as photon map evaluation. There are several applications:

Difficult Paths: Transport paths that are particularly hard to find randomly lead to large variance, because they may be found and give a high contribution, or not found and give no contribution. Among our images, the caustic caused by light bouncing off the mirror and through the glass ball in the Box scene of Figure 5 best fits this description. Light shining through a keyhole is perhaps the most commonly thought of example, if not the most common in practice. A user can supply paths that meet the geometric constraints and thus ensure the feature is adequately sampled.

User Control of Variance: Some regions of an image may be more important than others, such as those toward the center or in some other perceptually important region. A user can supply paths leading to the region of interest and it will be sampled with lower variance than other regions (Figure 3).

Resampling: Rather than a user defining paths, they could be taken from some previous sampling operation. Our earliest experiments used paths taken from the initial path tracing pass to estimate B_{eye}. Alternatively, a user could identify paths from a coarse run of the algorithm and reuse them in a final render. Resampling should also enable adaptive, unbiased Monte Carlo rendering and provide a handle on low-variance, physically-accurate animation rendering, but we leave these topics for future work.

Figure 3 compares image rendered with the Metropolis Light Transport algorithm: one with user paths and one without. Each image used 3 million iterations, producing a variance measurement of $VAR(E) = 1.04$ (Section 6) for the image with user input. It requires 4.8 million samples, or about 60% more time, to achieve similar results without the user input.

Figure 3: *An example of variance control due to the user path proposal strategy. Top is the image rendered with no user paths, while center is the result when the user specified ten paths passing through the doorway. Bottom are zooms of the wall intersection and table regions, with no user paths on the left and user paths on the right. These are MLT images that directly visualize the sampled light paths. The improvements after a final gather, while present, are less apparent.*

Reducing variance in one area of the image may lead to increased variance elsewhere, but it is not a zero-sum game. User paths can lead to a global reduction in variance if they increase the average acceptance probability, and hence the number of different paths sampled. This was the case in Figure 3, where the acceptance rate rose from 58% to 65% with

the introduction of user paths. In any event, users can choose to make a trade-off based on their own situation.

The user path proposal is not essential to achieving good results with Metropolis-Hastings sampling. It is a way to enhance control of the algorithm. The image in Figure 4 did not use the proposal, and the result in Figure 1 is almost as good without the user paths.

5.1. Candidates from User Paths

Each path provided by the user must start at a light and end at a diffuse surface. To obtain paths, we built a simple interface for the Box scene which allowed a user to interactively vary the origins and directions of rays from the light which were then traced through the scene and extracted as user paths. Tools like this could readily be included in modeling packages. For Figure 3 we specified paths by hand based on knowledge of the geometry.

Each path is input to the system as a sequence of surface points at which scattering occurs. These are stored as a set, $\{\overline{\mathbf{u}}_1,\ldots,\overline{\mathbf{u}}_{N_{UP}}\}$, containing N_{UP} paths. The first step of a proposal is to choose, uniformly at random, one of the input paths, $\overline{\mathbf{u}} = \langle \mathbf{x}_0,\ldots,\mathbf{x}_m\rangle$. This path forms a skeleton that we perturb to form the candidate path. The perturbation explores the space around the user path while avoiding the accumulation of a large power at a single photon.

The candidate path, $\langle \mathbf{x}'_0,\ldots,\mathbf{x}'_m\rangle$, is built starting at the light: $\mathbf{x}'_0 = \mathbf{x}_0$. We randomly generate a direction within a cone about axis $\mathbf{x}_0 \rightarrow \mathbf{x}_1$ by sampling θ, the angle between the axis and the direction, uniform in $[0,\beta)$ and ϕ, the azimuthal angle, uniform in $[0,2\pi)$. The surface point struck in this direction, $\mathbf{x}'_1$, is the next point on the candidate path. We repeat the process m times, using the direction $\mathbf{x}'_{i-1} \rightarrow \mathbf{x}_i$ as the axis of the sample cone. To form a complete path to the eye, the sub-path of sample X_{t-1} joining the eye to the first diffuse hit point is appended to the candidate. The candidate is rejected if there is no such diffuse point. When setting β, lower values are good for exploring tightly constrained paths while higher values give more variation around the user path and hence reduce variance over a larger area. The user can also specify a different β for each path segment.

The candidate path may pass through an opaque surface, in which case a visibility term in f_{eye} is zero and the path will be rejected. If the user path contains specular interactions, a specular surface must be found at the same index on the perturbed path. If it is, we follow the specular bounce rather than sampling a perturbed direction. If the user path specularity is not matched in the perturbed path, or the perturbed path intersects an unmatched specular surface, the candidate is rejected. These restrictions ensure that the specular bounces "cancel out" in computing the acceptance probability (see Veach [Vea97, §10.3.5]).

5.2. User Path Transition Functions

The transition probability must consider all the possible ways a UP proposal may have generated the candidate:

$$T_{UP}(X'_t|X_{t-1}) = \frac{1}{N_{UP}}\sum_{i=1}^{N_{UP}} C(\overline{\mathbf{u}}_i)\prod_{j=0}^{m-1} p_j \frac{G(\mathbf{x}'_j \leftrightarrow \mathbf{x}_{j+1})}{\cos\theta'_j} \quad (6)$$

$C(\overline{\mathbf{u}}_i)$ is 1 if the candidate could have been generated from path $\overline{\mathbf{u}}_i$, otherwise 0. The product of terms accounts for the probability of each perturbed bounce. If the bounce at $\mathbf{x}_j$ was non-specular, then $p_j = 1/2\pi\beta_j$. For a specular bounce, $p_j = 1$ because there is no random choice. The geometry terms are still required to convert from solid angle measure to surface area measure. The geometry and cosine term convert the direction sampled according to solid angle measure into one sampled using the surface area measure. θ'_j is the angle between the normal at $\mathbf{x}'_j$ and the direction $\mathbf{x}'_j \rightarrow \mathbf{x}_{j+1}$.

To compute $C(\overline{\mathbf{u}}_i)$, we perform the procedure for building a candidate from $\overline{\mathbf{u}}_i$, but rather than creating the new candidate we check that the point $\mathbf{x}_0$ is common to X'_t and $\overline{\mathbf{u}}_i$ and that each ray direction in X'_t lies within the sample cone of $\overline{\mathbf{u}}_i$. Finally, the resulting number of path segments must correspond. The reverse transition probability, $T_{UP}(X_{t-1}|X'_t)$, is is similarly computed.

The UP proposal generates a path, X'_t, close to a user given path regardless of the previous path, X_{t-1}. However, in most cases the path X_{t-1} could not have been generated from X'_t in the same manner; most paths are not close to user defined. Hence, $T_{UP}(X_{t-1}|X'_t)$ will be zero in almost all cases. This leads to a zero acceptance probability, which is a problem because the proposed path will never be used. It is, however, possible to generate a UP proposal candidate using a BD proposal because the latter gives any path a non-zero transition probability. Hence, we combine the UP and BD proposal strategies when computing transition functions: if *chosen* is either *UP* or *BD*, then

$$T(X'_t|X_{t-1}) = \frac{P_{UP}T_{UP}(X'_t|X_{t-1}) + P_{BD}T_{BD}(X'_t|X_{t-1})}{P_{UP}+P_{BD}} \quad (7)$$

Thus we have a two tiered proposal selection process. First, we decide if the proposal will be a UP-BD hybrid (with probability $P_{UP}+P_{BD}$) or one of the others. We apply Equation 5 for this selection. If the hybrid is chosen, we decide between UP and BD, and apply Equation 7.

The combination of UP and BD proposals in computing the transition functions is the key idea for enabling user input samples, and is possible because the acceptance probability mechanism of a Metropolis-Hastings sampler allows different sampling processes (proposal strategies) to be combined. Furthermore, the acceptance criteria ensures that the final distribution is unbiased provided the transition functions and target pdf values are correctly computed. Intuitively, the algorithm rejects just the right proportion of UP candidates to ensure that the final result is not biased toward them.

The values for P_{UP} and P_{BD} will influence performance of the algorithm. Assume that the reverse transition function, $T_{UP}(X'_{t-1}|X_t)$, is very small or zero and consider P_{UP}/P_{BD}, the ratio of UP to BD proposals. As P_{UP}/P_{BD} increases, the acceptance probability (Equation 3) will decrease, resulting in the chain repeating the same path more often. This results in fewer photons stored away from the user path (fewer candidates for these paths are proposed), but increases the power of those photons, resulting in a noisier image away from the user path. This effect is counter-balanced by the ratio of the $f_{eye,Y}$ terms, which favors transitions to important paths, including user paths, regardless of how they were proposed.

When using user paths to overcome hard-to-find paths, the ratio P_{UP}/P_{BD} should be higher to provide many user candidates which will be accepted due to their high $f_{eye,Y}$. In the context of user-guided variance reduction, the ratio should be smaller to avoid frequent rejection of user path candidates and the higher variance that would result in regions away from the user paths. Varying the ratio gives the user control over how much influence their paths have on the distribution of variance over image.

Rather than users providing paths, the user-path proposal could be extended to include hints about important surface patches or reflectance directions. To use important surface patches, for instance, the candidate path would be constructed by randomly choosing points on the patches and joining them up. The terms inside the product in Equation 6 must be modified to account for the new probabilities of choosing the points. Otherwise the algorithm is unchanged.

5.3. Photon Map Proposal

The PM proposal generates complete paths with eye subpaths that are similar to those used in the final gather phase. Photons derived from the complete paths will thus be at locations later useful for gathering. Tracing back toward the light from the last diffuse surface point, $\mathbf{x}_d$, (that closest to the eye) we find a sub-path $\langle \mathbf{x}_{d-k}, \ldots, \mathbf{x}_d \rangle$ of the form $(L|D)DS^*D$. That is, the sub-path back through any number of specular bounces (possibly 0) followed by a diffuse bounce and ending at the next diffuse surface, or the light. The candidate path keeps $\mathbf{x}_d$ and modifies the direction back to $\mathbf{x}_{d-1}$, similar to the way a final gather operation distributes rays to estimate indirect illumination.

We modify the central DS^* portion of the sequence by perturbing the direction of the ray $\mathbf{x}_d \rightarrow \mathbf{x}_{d-1}$ by an angle θ uniform in $[0,\gamma)$ and ϕ uniform in $[0,2\pi)$ (as in the UP proposal). For all examples in the paper we set $\gamma = 30^\circ$, and the precise value seems not to impact the results. This ray is traced back through zero or more specular bounces until the next diffuse hit, forming a new DS^* sequence which is inserted in place of the original, resulting in $\langle \mathbf{x}_{d-k}, \mathbf{x}'_{d-k'-1}, \ldots, \mathbf{x}'_{d-1}, \mathbf{x}_d \rangle$. The diffuse (or light) points on the end of the modified segment allow for non-zero probability that the candidate path will carry some power.

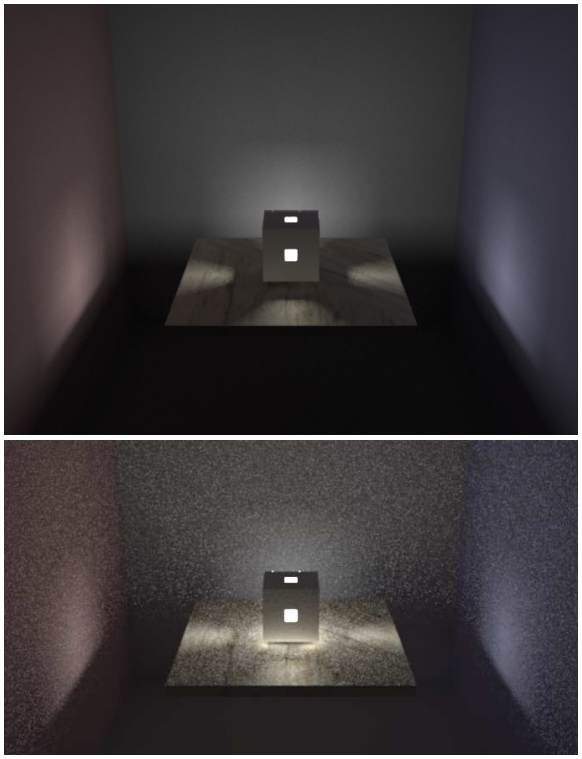

Figure 4: *A Jack-o-Lantern scene demonstrating MPS's efficient placement of samples. The Photon Mapping scene (lower) stores excess photons inside the box and an insufficient number on the walls of the room, resulting, respectively, in significant energy bleeding around the base of the box on the table and noise throughout the image.*

The transition probability is similar to that of the UP proposal, except that there is only one perturbed choice followed by a number of specular bounces:

$$T_{PM}(X'_t|X_{t-1}) = \frac{G(\mathbf{x}_d, \mathbf{x}_{d-1})}{2\pi\gamma\cos\theta_d} \prod_{j=d-1}^{d-k'-2} \frac{G(\mathbf{x}'_j, \mathbf{x}'_{j+1})}{\cos\theta'_j}$$

6. Results and Discussion

Our rendering system uses libraries and code from the PBRT toolkit [PH04] wherever possible, including for the final gather operation. There are a variety of parameters to the algorithm. Those for the MLT-style proposals were taken from Veach [Vea97]. Of the Photon Mapping final gather parameters, the formula for computing the maximum search distance for photons, d_{max}, was taken from Suykens [Suy02, Pg. 159] (with the parameter $\alpha = 0.1$) while the maximum number of photons in an estimate, n, was set at 60. We introduced new parameters for the probability of choosing a proposal strategy, P_{type}, which are given below on a per-image basis. We also introduced parameters for controlling the perturbation of a user path, β, which we varied per image, and the perturbation of a photon map sub-path, $\gamma = 30^\circ$.

Scene	Resolution	t_B (s)	t_{map} (s)		t_{FG} (s)		Total t (s)		# Photons		RMS Error	
			MPS	PM	MPS	PM	MPS	PM	MPS	PM	MPS	PM
Rooms	720×405	21	40	9	419	469	480	478	81004	300000	0.036	0.4239
Lantern	684×513	11	10	4	185	198	206	202	8675	37160	0.0728	1.165
Box	640×480	9	26	12	208	230	243	242	47798	250000	0.0214	0.0227

Table 1: *Statistics for images in the paper. Timing is given for MPS and Photon Mapping: t_B is the time to estimate B_{eye}, t_{map} is the photon sampling time and t_{FG} is final gather time. While MPS spends more time sampling, the fewer, well-distributed photons reduces the time required for the final gather. We also give the number of photons stored. Memory usage for the maps is linear in photon number, with 49 bytes per photon in the PBRT implementation that we use [PH04]. Finally, we give RMS errors for the images compared against path tracing solutions that ran for several days (Figure 6).*

Timing results and other statistics for the images in the paper are provided in Table 1. All images for comparison between methods were generated with nearly equal total computation time. All were reconstructed with a Gaussian kernel of width 2 pixels and $\sigma = 1$. Irradiance caching [WH92] was used to speed up photon map estimation [Jen01]. For tone reproduction we used Reinhard et al. [RSSF02], with the parameter $y_{max} = 100$. We implemented one further optimization borrowed from MLT. Rather than storing nothing for rejected paths, we store a photon with power reduced according to the acceptance probability, and reduce the power of the repeated path to compensate [Vea97]. This increases the number of photons stored and extracts some benefit from rejected paths, but at the cost of increased variance in photon power. We have found the benefits of increased usable photons to outweigh the variance.

We also computed error measurements with respect to a long running path tracing estimate of each image. For each pixel, we computed the relative error (before tone mapping):

$$E(x,y) = \frac{I(x,y) - I_{ref}(x,y)}{I_{ref}(x,y)}$$

where I_{ref} is the pixel luminance value from the path tracing reference image. In Table 1, we report the RMS value of these errors over the entire image, for MPS sampling and standard photon map sampling. MPS out-performs PM in all cases (although by a negligible amount in the Box example). Note that we cannot expect zero error here – even the reference image contains noise.

The Rooms scene of Figure 1 contains about 42,000 primitives. Both the Photon Mapping and MPS images used 4 samples per pixel and 40 final gather rays per sample for estimating indirect illumination. The scene contained user paths specified by hand but no caustics, and we set $\beta = 5^\circ$ in the user path mutation. The proposal probabilities were: P_{UP}=0.1, P_{BD}=0.3, P_{PM}=0.2, P_{CP}=0, P_{LP}=0.2 and P_{LS}=0.2. These, like all our proposal probabilities, were chosen to give roughly equal proportion to each strategy that was useful for the scene. While MPS spent significantly more time than Photon Mapping in sampling photons, it was regained in the faster final gather phase; MPS's smaller number of well-distributed photons improved the performance of nearest neighbor searching in the photon map. We also rendered this scene with Photon Mapping using 6 million photons, which took almost an hour and reduced the noise in the result, but failed to remove the energy bleeding problems and used two orders of magnitude more memory than MPS.

Apart from managing difficult transport paths, a significant advantage of MPS is its ability to store photons only where relevant. Figure 4 demonstrates a scene in which Photon Mapping stores almost all photons inside the lantern, where they remain unused when gathering for wall pixels. In contrast, MPS places almost all samples on the walls of the room. This results in reduced energy bleeding on the table around the box and far less noise in the image overall. These images used 30 samples for each indirect illumination estimate, and 4 samples per pixel. This scene contained no user paths (the important transport paths are not too hard to sample) nor caustics, hence the proposal probabilities were: P_{UP}=0, P_{BD}=0.4, P_{PM}=0.2, P_{CP}=0, P_{LP}=0.2 and P_{LS}=0.2.

Figure 5 shows a variant on the Cornell Box scene with complex caustic paths (the right wall and rear ball are mirrors, and the front ball is glass). We used ten user paths in this scene, five for each caustic under the ball. These were perturbed using $\beta = 1^\circ$ for segments between the light and mirror wall, and $\beta = 5^\circ$ for segments from the light direct to the glass ball. We set P_{UP}=0.1, P_{BD}=0.3, P_{PM}=0.2, P_{CP}=0.12, P_{LP}=0.08 and P_{LS}=0.2. Photon Mapping requires many photons to resolve the small caustic due to light bouncing off the mirror through the glass ball. Furthermore, the mirror wall subtends a large area at the light, so it is difficult to concentrate photon sampling toward the caustic producing region, and caustic photons sparsely stored on the rear wall cause excess noise due to their high power. Even with more photons, the caustic is not as good as that from MPS.

6.1. Limitations and Extensions

MPS is slower per photon than standard Photon Mapping, but a greater proportion of the stored photons are typically useful. The increase in per-photon cost is because more terms must be evaluated to determine the acceptance probability for each candidate. A path tracing phase is also required and its cost should be amortized over the stored photons. However, the significant improvement in photon distribution achieved with MPS allows for fewer photons overall

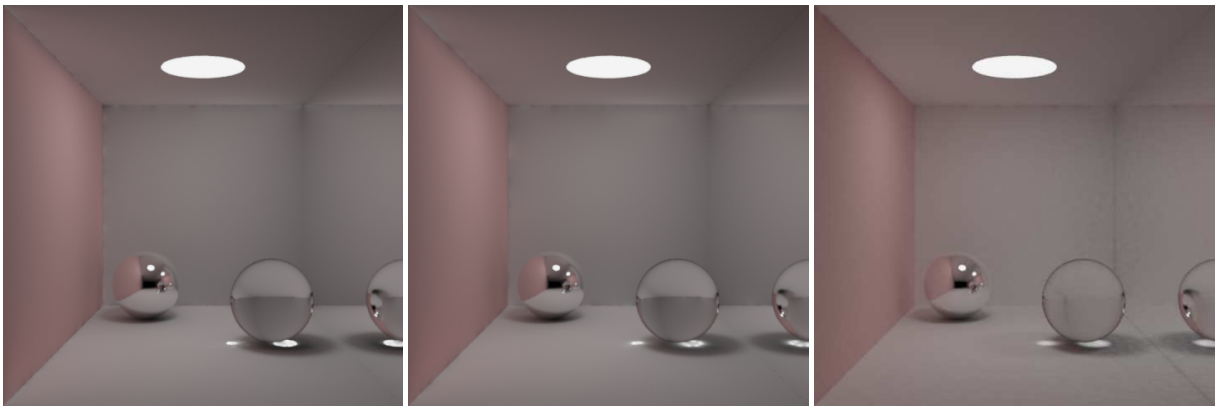

Figure 5: *The Box scene has a mirror ball at the rear and a mirror right wall, while the front ball is glass. The left image included ten paths specified by the user: five contribute to the large caustic under the glass ball, while the others bounce off the mirror and through the ball to contribute to the smaller caustic. The center scene had no user paths, and consequently the caustics show high variance. Right is a Photon Mapping image of the Box scene computed in equivalent time. The large number of photons cast to resolve the small caustic result in slightly greater noise in the right-rear of the box.*

Figure 6: *Reference images for the scenes in the paper, generated using path tracing.*

and typically reduces the cost of the final gather, giving better images for a given computational effort. We have also lost the view invariance of standard photon map construction, as would any method using visual importance. If the viewer's path were known, the eye location could be a variable included in the sampling process, just as locations on an area light source can vary.

Samples from a Metropolis-Hastings algorithm are correlated due to the Markov process, so the chain needs some time to explore the space adequately, whereas independent particles traced from the light will show no spatial correlation, and can be stratified across the light surface and outgoing direction. This may be important in scenes with very few photons. Parallel Markov chains could be used to generate samples, which would improve the distribution of samples over very short runs. We found this made no difference to the results for the photon counts required in our scenes.

Alternate methods could be used to sample paths, such as bi-directional path tracing or path tracing from the eye. These would be simpler to implement, and less computationally expensive, but lack the ability of MPS to exploit correlation in power between neighboring paths. A production system should support multiple algorithms for populating photon maps and share the final gather code and many other modules, including those for ray-tracing and BRDF sampling. Our system is built this way.

We store photons only at a single point along a sampled path — the point most relevant to a final gather operation. However, other points along the path may also be useful, as is the case in the Box scene where any diffuse surface point may be called upon to compute a radiance estimate. We choose not to store additional points because of the memory overhead and the energy bleeding problem. An alternative is to use an importon map to measure the visual importance of surface points, and store photons at any sufficiently im-

portant point along the path [KW00]. This would probably reduce the number of iterations required for MPS on simple scenes, at the cost of an importon map construction phase.

The target pdf we use, f_{eye}, considers all paths that carry power from the lights to the image as important. We could support other forms of importance, such as perceptual metrics or shading discontinuities, simply by modifying the $W_{eye}(\bar{\mathbf{x}})$ component of f_{eye}. The only potential downside would be an increase in the variability of power stored at the photons, $\Phi^{(j)}$, which can increase noise in the final image.

The user path proposal can be used, unmodified, for Metropolis Light Transport (Figure 3). Its impact is even greater because the variance in MLT is not disguised by the final gather operation. Conversely, MLT offers a variance reduction technique that we did not implement: the brightness of image pixels is estimated in a first pass and used to modify the path probabilities to make all pixels equally probable. This could be implemented in MPS through importon maps that modified the probability of paths, but it may result in large variance in photon powers.

Finally, our work could be extended to atmospheric scattering by combining Photon Mapping for participating media [JC98] with Pauly et al.'s [PKK00] MCMC sampler.

7. Conclusion

Metropolis Photon Sampling succeeds in generating photon map samples that meet the needs of the final gather phase, without wasting storage or computation time on unnecessary photons. It achieves this by sampling only over light transport paths that reach the image, and storing photons only at appropriate points along the path. The photon distribution that results has more photons that contribute to visually important locations, and fewer in irrelevant places. This not only improves estimates from the map due to higher photon density, but also reduces the chance that inappropriate photons will be used and hence reduces energy bleeding artifacts. At the same time, MPS allows users to supply information to the sampler in the form of important paths, something not achievable in most Monte Carlo algorithms.

The new sampler is best suited to scenes in which only a small portion of the lights' power arrives in visually important areas. Our method does not require any modification to the final gather phase of photon mapping, so it can be used in conjunction with a standard particle tracing sampler. Depending on the scene, one or other sampler could be used, but there is nothing preventing the use of both methods to fill the same map in scenes with multiple light sources that contribute differently to the image. Furthermore, any improvements to the final gather phase of Photon Mapping apply equally well to Metropolis Photon Sampling.

Acknowledgements

Thanks to Peter Shirley, Matt Pharr, David Forsyth and John Hughes for comments on early drafts of the paper. This work was funded by NSF grant CCR-0204372, with equipment donations from Intel.

References

[AD99] ANDRIEU C., DOUCET A.: Joint bayesian model selection and estimation of noisy sinusoids via reversible jump MCMC. *IEEE Transactions on Signal Processing 47*, 10 (1999), 2667–2676. 6

[Arv86] ARVO J.: Backward ray tracing. *Developments in Ray Tracing. ACM SIGGRAPH course notes 12* (1986), 259–263. 3

[Chr03] CHRISTENSEN P. H.: Adjoints and importance in rendering: An overview. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 1–12. 3

[Col94] COLLINS S.: Adaptive splatting for specular to diffuse light transport. In *Rendering Techniques '94 (Proceedings of the 5th Eurographics Workshop on Rendering)* (1994), pp. 119–135. 3

[CRMT91] CHEN S. E., RUSHMEIER H. E., MILLER G., TURNER D.: A progressive multi-pass method for global illumination. In *SIGGRAPH '91: Proceedings of the 18th annual conference on computer graphics and interactive techniques* (1991), pp. 165–174. 3

[DBMS02] DMITRIEV K., BRABEC S., MYSZKOWSKI K., SEIDEL H.-P.: Interactive global illumination using selective photon tracing. In *Rendering Techniques '02 (Proceedings of the 13th Eurographics Workshop on Rendering)* (2002), pp. 25–36. 3

[Dri01] DRIEMEYER T.: *Rendering with Mental Ray*, 2nd ed. Springer, 2001. 1

[GRS96] GILKS W. R., RICHARDSON S., SPIEGELHALTER D. J.: *Markov chain Monte Carlo in Practice*. Chapman & Hall, 1996. 2, 5

[Has70] HASTINGS W. K.: Monte Carlo sampling methods using Markov chains and their applications. *Biometrika 57* (1970), 97–109. 2

[Hec90] HECKBERT P. S.: Adaptive radiosity textures for bidirectional ray tracing. In *SIGGRAPH '90: Proceedings of the 17th annual conference on computer graphics and interactive techniques* (1990), pp. 145–154. 3

[ICG86] IMMEL D. S., COHEN M. F., GREENBERG D. P.: A radiosity method for non-diffuse environments. In *SIGGRAPH '86: Proceedings of the 13th annual conference on computer graphics and interactive techniques* (1986), pp. 133–142. 2

[JC98] JENSEN H. W., CHRISTENSEN P. H.: Efficient simulation of light transport in scenes with participating

media using photon maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on computer graphics and interactive techniques* (1998), pp. 311–320. 11

[Jen01] JENSEN H. W.: *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001. 1, 3, 9

[Kaj86] KAJIYA J. T.: The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on computer graphics and interactive techniques* (1986), pp. 143–150. 2

[KK00] KOLLIG T., KELLER A.: Efficient bidirectional path tracing by randomized quasi-monte carlo integration. In *Monte Carlo and Quasi-Monte Carlo Methods*, Fang K.-T., Hickernell F.,, Niederreiter H., (Eds.). Springer-Verlag, 2000, pp. 290–305. 2

[KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the Metropolis light transport algorithm. In *Computer Graphics Forum (Proceedings Eurographics 2002)* (2002), pp. 531–540. 2

[KW00] KELLER A., WALD I.: Efficient importance sampling techniques for the photon map. In *Proc. Vision, Modelling and Visualization 2000* (2000), pp. 271–279. 3, 11

[LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)* (1993), pp. 145–153. 2

[MRR*53] METROPOLIS N., ROSENBLUTH A. W., ROSENBLUTH M. N., TELLER A. H., TELLER E.: Equation of state calculations by fast computing machine. *The Journal of Chemical Physics 21*, 6 (1953), 1087–1092. 2

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering from Theory to Implementation*. Morgan Kaufmann, 2004. 2, 4, 8, 9

[PKK00] PAULY M., KOLLIG T., KELLER A.: Metropolis light transport for participating media. In *Rendering Techniques '00 (Proceedings of the 11th Eurographics Workshop on Rendering)* (2000), pp. 11–22. 3, 11

[PP98] PETER I., PIETREK G.: Importance driven construction of photon maps. In *Rendering Techniques '98 (Proceedings of the 9th Eurographics Workshop on Rendering)* (1998), pp. 269–280. 3

[RSSF02] REINHARD E., STARK M., SHIRLEY P., FERWERDA J.: Photographic tone reproduction for digital images. In *SIGGRAPH '02: Proceedings of the 29th annual conference on computer graphics and interactive techniques* (2002), pp. 267–276. 9

[SP89] SILLION F., PUECH C.: A general two-pass method integrating specular and diffuse reflection. In *SIGGRAPH '89: Proceedings of the 16th annual conference on computer graphics and interactive techniques* (1989), pp. 335–344. 2

[Suy02] SUYKENS F.: *On Robust Monte Carlo Algorithms for Multi-pass Global Illumination*. PhD thesis, Computer Science, K.U. Leuven, Belgium, 2002. 8

[SW00] SUYKENS F., WILLEMS Y. D.: Density control for photon maps. In *Rendering Techniques '00 (Proceedings of the 11th Eurographics Workshop on Rendering)* (2000), pp. 23–34. 3

[SWH*95] SHIRLEY P., WADE B., HUBBARD P., ZARESKI D., WALTER B., GREENBERG D.: Global illumination via density-estimation. In *Rendering Techniques '95 (Proceedings of the 6th Eurographics Workshop on Rendering)* (1995), pp. 219–230. 3

[Tie98] TIERNEY L.: A note on Metropolis-Hastings kernels for general state spaces. *The Annals of Applied Probability 8*, 1 (1998), 1–9. 6

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 4, 5, 7, 8, 9

[VG94] VEACH E., GUIBAS L. J.: Bidirectional estimators for light transport. In *Rendering Techniques '94 (Proceedings of the 5th Eurographics Workshop on Rendering)* (1994), pp. 147–162. 2

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on computer graphics and interactive techniques* (1995), pp. 419–428. 2

[VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *SIGGRAPH '97: Proceedings of the 24th annual conference on computer graphics and interactive techniques* (1997), pp. 65–76. 2, 5

[War94] WARD G. J.: The RADIANCE lighting simulation and rendering system. In *SIGGRAPH '94: Proceedings of the 21st annual conference on computer graphics and interactive techniques* (1994), pp. 459–472. 3

[WCG87] WALLACE J. R., COHEN M. F., GREENBERG D. P.: A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. In *SIGGRAPH '87: Proceedings of the 14th annual conference on computer graphics and interactive techniques* (1987), pp. 311–320. 2

[WH92] WARD G. J., HECKBERT P.: Irradiance gradients. In *Proceedings of the 3rd Eurographics Workshop on Rendering* (1992), pp. 85–98. 3, 9

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *SIGGRAPH '88: Proceedings of the 15th annual conference on computer graphics and interactive techniques* (1988), pp. 85–92. 3