

Robust Adaptive Photon Tracing Using Photon Path Visibility

TOSHIYA HACHISUKA and HENRIK WANN JENSEN

University of California, San Diego

We present a new adaptive photon tracing algorithm which can handle illumination settings that are considered difficult for photon tracing approaches such as outdoor scenes, close-ups of a small part of an illuminated region, and illumination coming through a small gap. The key contribution in our algorithm is the use of visibility of photon path as the importance function which ensures that our sampling algorithm focuses on paths that are visible from the given viewpoint. Our sampling algorithm builds on two recent developments in Markov chain Monte Carlo methods: adaptive Markov chain sampling and replica exchange. Using these techniques, each photon path is adaptively mutated and it explores the sampling space efficiently without being stuck at a local peak of the importance function. We have implemented this sampling approach in the progressive photon mapping algorithm which provides visibility information in a natural way when a photon path contributes to a measurement point. We demonstrate that the final algorithm is strikingly simple, yet effective at sampling photons under lighting conditions that would be difficult for existing Monte Carlo ray tracing-based methods.

Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—*Raytracing*

General Terms: Algorithms

Additional Key Words and Phrases: Global illumination, photon mapping, density estimation, adaptive sampling

ACM Reference Format:

Hachisuka, T. and Jensen, H. W. 2011. Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph.* 30, 5, Article 114 (October 2011), 11 pages.

DOI = 10.1145/2019627.2019633

<http://doi.acm.org/10.1145/2019627.2019633>

1. INTRODUCTION

Developing efficient and robust global illumination algorithms has been an active area of research in computer graphics over the last

Authors' addresses: T. Hachisuka (corresponding author) and H. W. Jensen, Department of Computer Science, University of California, San Diego, San Diego, CA; email: thachisu@cs.ucsd.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2011 ACM 0730-0301/2011/10-ART114 \$10.00

DOI 10.1145/2019627.2019633

<http://doi.acm.org/10.1145/2019627.2019633>

25 years. Current state-of-the-art algorithms are based on Monte Carlo ray tracing. These algorithms solve the rendering equation [Kajiya 1986] and handle complex geometries, materials, and lighting conditions. However, there are scene configurations where many of the Monte Carlo ray tracing algorithms become highly inefficient. An example would be the interior of a room illuminated by sunlight through the window. If the exterior scene geometry is relatively large, efficiently generating light paths through the window that contribute to the rendered image becomes challenging. Another example would be detailed caustics patterns from a headlight of a car or strong indirect illumination coming from an adjacent room through a small gap.

To address difficult lighting scenarios, a number of improved sampling algorithms have been developed. The most successful ones are based on Markov chain Monte Carlo methods which were introduced by Veach and Guibas [1997] in computer graphics, who developed the Metropolis light transport method. Metropolis light transport improves the efficiency of light path sampling by generating a new light path by a small perturbation of the previous light path. Unfortunately, unbiased Monte Carlo ray tracing methods, including Metropolis light transport, become highly inefficient for specular-diffuse-specular light paths from small light sources [Veach 1998; Hachisuka et al. 2008]. This is inconvenient as most natural scenes with light bulbs or sunlight exhibit such light paths. An example includes rendering of a glass illuminated by sunlight with a diffuse surface below the glass.

Recently, Hachisuka et al. [2008] presented progressive photon mapping as a robust alternative to unbiased Monte Carlo ray tracing methods. Progressive photon mapping can handle specular-diffuse-specular light transport robustly, however, it becomes inefficient in scenes where only a small part of the lit surfaces can be seen in the rendered image. In general, this type of scene is problematic for any photon-tracing-based methods including progressive photon mapping and the original photon mapping algorithm [Jensen 1996]. To address this issue, Fan et al. [2005] proposed a Markov chain sampling method to improve photon tracing in such scenarios. Unfortunately, their method uses path tracing from the eye to seed the sampling process, and it is still inefficient for specular-diffuse-specular light transport similarly to unbiased Monte Carlo ray tracing methods.

In this article, we propose a simple, automatic, and robust photon tracing algorithm that extends the types of scenes that can be rendered efficiently with photon tracing-based methods. The key idea is a new importance sampling function solely based on the visibility information of each photon path. In order to generate samples from this importance function, we apply two recent developments in Markov chain Monte Carlo methods: adaptive Markov chain sampling and replica exchange. Adaptive Markov chain sampling adjusts the parameters of the mutation strategies adaptively and removes the need for a user to tweak sampling parameters. Applying replica exchange to progressive photon mapping provides an automatic mixture of uniform random samples and Markov chains, and ensures that photon paths do not get stuck at local peaks of the importance function. Our method is effective in many cases and strikingly simple to implement. Figure 1 highlights the results of

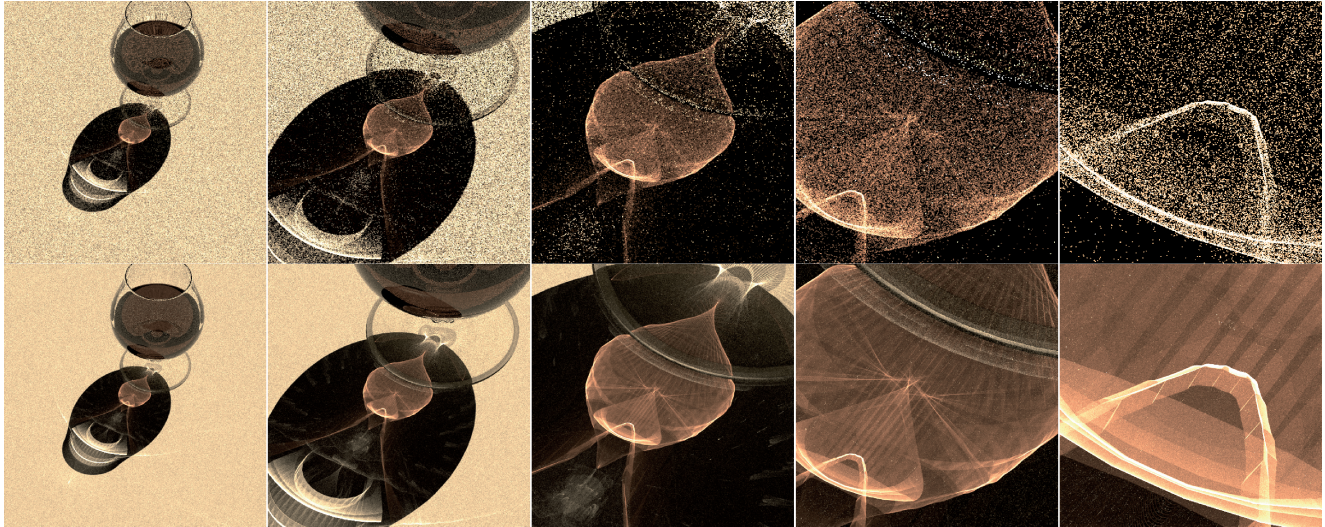


Fig. 1. Cognac glass illuminated by a directional light source. The figure compares rendered images using progressive photon mapping with the same rendering time (120 min), but with different photon tracing algorithms. The images on the top row are rendered using random sampling of photons, which become increasingly noisy as we zoom into the caustic. Using our photon tracing method (bottom row), we can focus tracing photon paths into the region that contributes to the image without any portal, and render the close-ups with less noise in the same rendering time. Note that no other existing global illumination methods can render illumination under the cognac glass accurately, since this illumination comes from specular-diffuse-specular paths from a light source with zero solid angle. The combination of progressive photon mapping and our photon tracing technique is the first method that works effectively and robustly in this kind of scene. The stripe patterns in the caustic are not artifacts of our method; they are caused by the tessellation of the cognac glass.

our algorithm on a close-up of a caustic pattern created by a cognac glass. In summary, our contributions are the following.

- Use of visibility as an importance function in path space, which significantly simplifies the implementation.
- Introduction of adaptive Markov chain Monte Carlo methods to rendering, which makes the algorithm parameter-free.
- Application of replica exchange Monte Carlo method, which provides an automatic mixture of uniform random samples and Markov chains.

2. RELATED WORK

Many rendering algorithms use photon tracing in various forms for solving the light transport problem [Arvo 1986; Dutré et al. 1993; Jensen 1996; Lafortune and Willems 1993; Veach and Guibas 1995]. Most recently, Hachisuka et al. [2008] proposed progressive photon mapping which solves the light transport problem by using progressive accumulation of photons. Our method uses progressive photon mapping as its rendering algorithm since it can robustly render scenes that are difficult to render with other methods (e.g., reflections of caustics from a directional light source).

Photon tracing methods can, however, become inefficient when only a small part of the illuminated scene is visible. Veach and Guibas [1997] addressed a related problem by introducing the Metropolis light transport method as an application of Markov chain Monte Carlo methods to rendering. Their assumption is that, given a sampled light path that we already know contributes to the image, similar light paths will be likely to contribute to the image as well. Under this assumption, Metropolis light transport generates a new path by slightly perturbing the previous path. They demonstrated that this strategy works well in difficult settings including

illumination coming through a small gap. Markov chain Monte Carlo methods have been successfully used in other forms as well such as multiple-try Markov chain Monte Carlo methods [Segovia et al. 2007] and energy redistribution path tracing [Cline et al. 2005].

Most related to our work is the method by Fan et al. [2005], who applied Metropolis light transport to photon tracing. In order to obtain the full information of a photon path including whether it contributes to the image, they proposed to directly sample a complete path from the path space that connects a light source and the eye. Common to existing MCMC rendering methods and their work, however, is that they all use sampling on the exact path space. This unfortunately becomes inefficient in the presence of some light paths, such as specular-diffuse-specular paths from small light sources. This is because such paths will have nearly zero volume (or zero in the case of point light sources or directional light sources) in the path space [Veach 1998]. These paths are rather common in the real world such as illumination coming from the filament in a light bulb. Our method avoids this issue by sampling a blurred path space using progressive photon mapping, where blurring vanishes as we add more samples.

Our algorithm uses two advancements in Monte Carlo sampling algorithms: adaptive Markov chain Monte Carlo methods [Haario et al. 2001] and replica exchange [Swendsen and Wang 1986]. It has been known that the mutation parameters, thereby mutation kernels affect the performance of Markov chain Monte Carlo methods significantly, which are usually tuned by users. Adaptive Markov chain Monte Carlo methods automate this process by using information obtained from past samples and self-learning the importance function. In this article, we introduce adaptive Markov chain Monte Carlo methods to rendering. As far as we know, our method is the first application of adaptive Markov chain Monte Carlo methods in rendering.

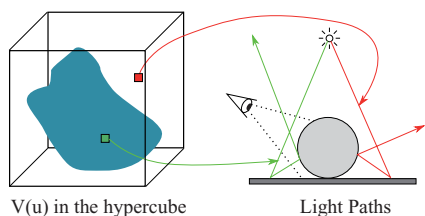


Fig. 2. Sampling space of our method. We define a function $V(x)$ in the hypercube of random numbers. The function returns 1 if a corresponding photon path contributes to the image (the green point in the shaded region) and 0 otherwise (the red point outside the shaded region).

The replica exchange method uses multiple target distributions, or an extended space of the target distribution with auxiliary parameters, in order to introduce interdistribution mutations. This alleviates the problem of a Markov chain being trapped within a single mode in a multimodal distribution by taking a “detour” of the Markov chain through different distributions. Kitaoka et al. [2009] applied this algorithm to modify the original Metropolis light transport method by defining multiple target distributions based on a heuristic separation of light paths (e.g., direct illumination, indirect illumination, caustics etc.). We provide a formulation of replica exchange Monte Carlo method in the context of our visibility function. The key difference is that our formulation results in a strikingly simple algorithm that is independent from scene settings.

3. METHOD

3.1 Overview

The overall idea of our algorithm is to define a visibility function of photon paths and to perform importance sampling on this visibility function. We define the space of this function as a hypercube similar to the one that was proposed by Kelemen et al. [2002]. Note that a point in this space corresponds to a set of random numbers. We then employ local importance sampling for choosing light sources and sampling BRDFs and Russian roulette, in order to generate a photon path from given random numbers. In order to efficiently sample this function, we propose a combination of adaptive Markov chain sampling and replica exchange as we will describe in the next sections.

3.2 Sampling Space and Visibility Function

We first define our sampling space and the importance function. Given a photon path, \vec{u} , in the hypercube, we define a *photon path visibility function*, $V(\vec{u})$, where $V(\vec{u}) = 1$ if any photon due to this photon path contributes to the image and $V(\vec{u}) = 0$ otherwise. The importance function is simply the normalized version of this visibility function $F(\vec{u})$, which is $F(\vec{u}) = \frac{V(\vec{u})}{V_c}$ where $V_c = \int V(\vec{u})d\vec{u}$. Figure 2 illustrates the definition of our sampling space and visibility function. The use of this function has the additional advantage that there is no local peak in the function, which is prone to high autocorrelation of samples in Markov chain Monte Carlo methods (e.g., a chain gets stuck in a very bright path). We can also easily evaluate $V(\vec{u})$ by checking if a photon path splats any photon into any of measurement points in the photon splatting implementation of progressive photon mapping that we describe next.

3.3 Photon Splatting Implementation

In this variation of progressive photon mapping, we construct an acceleration data structure of measurement points, not a photon map. In the succeeding photon passes, instead of storing photons as a photon map, we perform a range query over measurement points at each photon’s position. In other words, this algorithm is *splatting* photons into the measurement points, instead of *gathering* photons at each measurement point. We then apply the radius reduction and the flux correction normally to all of the affected measurement points. The radiance estimation at each measurement point can be done as usual. Resampling of measurement points in stochastic progressive photon mapping is done after tracing a user-specified number of photons, which controls the frequency of eye ray tracing. We use this splatting implementation throughout the article, in order to immediately utilize the visibility information of the current photon path to the next photon tracing.

4. REPLICA EXCHANGE MONTE CARLO

4.1 Overview

The replica exchange Monte Carlo method is an extended ensemble Monte Carlo method where we sample Markov chains from multiple distributions simultaneously (refer to Iba [2001] for an overview of this class of algorithms). The basic idea is facilitating exploration of the sampling space by bridging multiple distant peaks using another smooth importance function. For example, if we use a regular Markov chain Monte Carlo method to sample from an importance function with two peaks separated by zeros, the Markov chain can get trapped within one peak for many iterations. The replica exchange Monte Carlo method can avoid this problem by introducing an extra Markov chain, for instance, from a uniform distribution. Even if a Markov chain gets stuck in one peak, this chain can be exchanged with another Markov chain in the uniform distribution without changing the resulting sample distribution using replica exchange. Figure 4 illustrates this idea. We first describe a general formulation of replica exchange Monte Carlo method in the following and explain our formulation in the next.

Given a set of multiple importance functions, $F_1(\vec{u}), \dots, F_Q(\vec{u})$, we define a set of independently generated samples from each function as $\vec{U} = \{\vec{u}_1, \dots, \vec{u}_Q\}$. Under these definitions, \vec{U} can be considered a single sample from the following product function

$$\tilde{F}(\vec{U}) = \prod_{k=1}^Q F_k(\vec{u}_k), \quad (1)$$

where \vec{u}_k is a sample (or a state of the Markov chain) in the importance function F_k .

The key idea of the replica exchange Monte Carlo method is to perform an interdistribution exchange such that the preceding product distribution of samples remains unchanged. This can be achieved by exchanging states of two chains, \vec{u}_i and \vec{u}_j , with the probability

$$r(\vec{u}_i, \vec{u}_j) = \min\left(1, \frac{F_i(\vec{u}_j)F_j(\vec{u}_i)}{F_i(\vec{u}_i)F_j(\vec{u}_j)}\right). \quad (2)$$

As a result, each sequence of Markov chain \vec{u}_k still distributes according to $F_k(\vec{u})$, but possibly with a better exploration of the sampling space due to interdistribution exchanges. This formulation subsumes the large step mutation used by Kelemen et al. [2002] with a theoretical formulation using the product distribution.

Kitaoka et al. [2009] has applied the replica exchange Monte Carlo method to the light transport problem in the context of improving Metropolis light transport, where they defined target distributions of separated light paths in the heuristic order of difficulty. This separation is highly scene dependent and its implementation becomes relatively complex compared to the regular Metropolis light transport algorithm. Our algorithm is far simpler than their work in terms of implementation and independent from scene configurations.

4.2 Our Formulation

Although the replica exchange Monte Carlo method can use multiple importance functions, we only use two functions: the target importance function $F(\vec{u})$ as defined by the visibility function and a constant function $I(\vec{u}) = 1$. Note that even though $F(\vec{u})$ only returns either zero or $1/V_c$ without any local peak, samples in standard Markov chain Monte Carlo methods can still be trapped within one region in the hypercube for a large number of iterations (e.g., only sampling light from one window out of two windows in a room).

We therefore consider two Markov chains \vec{u}_F and \vec{u}_I from $F(\vec{u})$ and $I(\vec{u})$. Using Eq. (2), we exchange those two chains across the distributions with probability $r(\vec{u}_I, \vec{u}_F)$.

$$r(\vec{u}_I, \vec{u}_F) = \frac{F(\vec{u}_I)I(\vec{u}_F)}{F(\vec{u}_F)I(\vec{u}_I)} \quad (3)$$

For general importance functions, this equation needs computation of multiple probability density at arbitrary sample locations as in Eq. (2). However, we can simplify the equation in our method since we know that the sample \vec{u}_F always returns $F(\vec{u}_F) = \frac{1}{V_c}$ by definition and $I(\vec{u}_I) = I(\vec{u}_F) = 1$.

$$r(\vec{u}_I, \vec{u}_F) = \frac{F(\vec{u}_I)1}{\frac{1}{V_c}1} = V(\vec{u}_I) \quad (4)$$

The end result is straightforward. Since the sample \vec{u}_I is from uniform sampling, there is no need to keep track of a Markov chain of \vec{u}_I ; if uniform independent sampling generates a useful path (when $r(\vec{u}_I, \vec{u}_F) = V(\vec{u}_I) = 1$) we replace the current photon path, otherwise, we keep the current photon path and mutate normally (when $r(\vec{u}_I, \vec{u}_F) = V(\vec{u}_I) = 0$). This formulation results in an automatic mixture of uniform sampling and Markov chain sampling based on how frequently we obtain $V(\vec{u}_I) = 1$.

4.3 Progressive Estimation of the Normalization Term

Using the uniform distribution in the formulation of replica exchange Monte Carlo method gives us another benefit. By counting the samples from the uniform distribution, we can compute the normalization term V_c in a progressive fashion which is usually computed in a separate pass in existing Markov chain Monte Carlo rendering methods. We can estimate the normalization term as

$$V_c = \int V(\vec{u})d\vec{u} \approx \frac{N_{I,V(\vec{u})=1}}{N_{I,\text{total}}}, \quad (5)$$

where $N_{I,V(\vec{u})=1}$ is the number of visible paths from uniform distribution, and $N_{I,\text{total}}$ is the total number of generated paths from a uniform distribution. Note that $N_{I,\text{total}}$ is in fact equal to the total number of generated photon paths from $F(\vec{u})$ because we always generate a sample from the uniform distribution (Figure 3). Since this value is already kept for the purpose of radiance computation, the preceding computation of V_c requires keeping only one

```

MutationSize ← 1, Accepted ← 1, Mutated ← 0, UniformCount ← 1
repeat
  CurrentPath ← UNIFORM()
until ISVISIBLE(CurrentPath)
for i ← 1 to NumTotalPhotons
  UniformPath ← UNIFORM()
  if ISVISIBLE(UniformPath)
    then {
      CurrentPath ← UniformPath
      UniformCount ← UniformCount + 1
      CandidatePath ← MUTATE(CurrentPath, MutationSize)
      Mutated ← Mutated + 1
      if ISVISIBLE(CandidatePath)
        then {
          CurrentPath ← CandidatePath
          Accepted ← Accepted + 1
          R ← Accepted/Mutated
          MutationSize ← MutationSize + (R - 0.234) / Mutated
        }
    }
  else {
    then {
      CurrentPath ← CandidatePath
      Accepted ← Accepted + 1
      R ← Accepted/Mutated
      MutationSize ← MutationSize + (R - 0.234) / Mutated
    }
  }
SPLAT(CurrentPath)
DISPLAY(UniformCount/i)

```

Fig. 3. Our photon tracing algorithm. UNIFORM() samples a hypercube using uniform random numbers, and MUTATE() returns a mutated path with a given mutation strategy parameter (MutationSize). SPLAT() finds nearby measurement points of each photon and accumulates photon statistics. ISVISIBLE() returns true if the given photon path splats any photon into any of measurement points and false otherwise. DISPLAY() takes its argument as a scaling factor, and computes pixel values using current photon statistics.

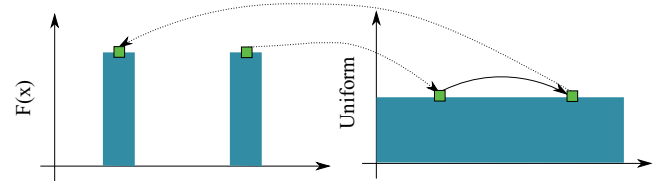


Fig. 4. Example of the replica exchange Monte Carlo method with two distributions. The target distribution (left), $F(x)$, can have multiple distant peaks that are difficult to sample with regular Markov chain Monte Carlo methods. The replica exchange Monte Carlo method can improve exploration of a Markov chain by combining a uniform distribution with interdistribution exchanges (right).

additional value, $N_{I,V(\vec{u})=1}$. This normalization constant, V_c , just uniformly scales the radiance estimate computed for each pixel.

5. ADAPTIVE MARKOV CHAIN MONTE CARLO

5.1 Overview

One notable difficulty in Markov chain Monte Carlo methods is that the optimal mutation strategy is problem dependent. For example, if we render an object on a plane with a point light source, depending on the relative size of the plane and the object, ranges of photon paths that intersect with the object and/or the plane in the hypercube change dramatically. In general, no single preset of mutation strategies will work well in all scene settings. We could have as many mutation strategies as possible to hope that at least one of them is effective, but this approach wastes computation on other ineffective mutations. Adaptive Markov chain Monte Carlo methods provide a way to automatically adjust mutation strategies during the computation by learning the importance function as we sample. Since

adaptive Markov chain Monte Carlo methods in general cover many different variations, we only provide an overview of the method that we use, which is a *controlled Markov chain Monte Carlo method* [Andrieu and Robert 2001]. For a more comprehensive overview of other methods, readers can refer to a survey such as the one by Andrieu and Thoms [2008].

The idea of a controlled Markov chain Monte Carlo method is to adjust the parameters of given fixed mutation strategies based on the previous samples. Given a vector of the initial parameter values, $\bar{\theta}_1$, a controlled Markov chain Monte Carlo method updates the parameters values $\bar{\theta}_i$ as

$$\bar{\theta}_{i+1} = \bar{\theta}_i + H(i, \bar{\theta}_i, \bar{u}_i, \dots, \bar{u}_1), \quad (6)$$

where $\bar{u}_i, \dots, \bar{u}_1$ are all samples up to the i th iteration and H is a function that computes the changes of the parameters according to this history of samples and the last parameter values $\bar{\theta}_i$. One important condition that H needs to satisfy in order to keep the sample distribution intact is *diminishing adaptation principle* [Andrieu and Thoms 2008].

$$\lim_{i \rightarrow \infty} H(i, \bar{\theta}_i, \bar{u}_i, \dots, \bar{u}_1) = 0 \quad (7)$$

There are many possible approaches to adapt the parameters while satisfying this condition, but one simple approach that is used in existing adaptive Markov chain Monte Carlo methods is changing the parameters such that an *acceptance ratio* of Markov chains reaches the desired value. The acceptance ratio (or *acceptance rate*) is the fraction of accepted mutations over all the mutations. For separable functions, the optimal asymptotic acceptance ratio has been derived 23.4% [Roberts et al. 1997]. We can thus simplify Eq. (6) as

$$\bar{\theta}_{i+1} = \bar{\theta}_i + H(i, A^*, A_i), \quad (8)$$

where A^* is the target acceptance ratio and A_i is the acceptance ratio of samples up to i .

While it is true that our importance function will not be separable in many scenes, previous work confirmed that using 23.4% works well in practice with nonseparable functions [Rosenthal et al. 2008]. Furthermore, the general principle that the acceptance ratio should not be too high or too low is applicable to any functions, so any target acceptance ratio that is not too close to 0% or 100% will work as we will demonstrate in the results section. We therefore use $A^* = 23.4\%$ in all the examples that we show in this article. The key is that the same target acceptance ratio will work well for many settings and the user does not need to tweak the target acceptance ratio scene by scene as we will demonstrate.

5.2 Our Formulation

In this article, we use a simple form of a controlled Markov chain Monte Carlo method, which adjusts a single mutation parameter in a power function. A mutation of each coordinate of a given point is done by adding

$$\Delta u = \text{sgn}(2\xi_0 - 1)\xi_1^{\frac{1}{\theta_i} + 1} \quad (9)$$

to each coordinate while keeping the result within (0, 1) by wrapping around the value in this range. θ_i is the adaptive mutation size at the i th Markov chain, $\text{sgn}(x)$ is a function that returns the sign of x , and ξ_0 and ξ_1 are uniform random numbers within (0, 1). The mutation size is a global value that is maintained throughout the sampling process. Note that $\theta_i = \infty$ corresponds to uniform random sampling which generates as large mutation as possible, and $\theta_i = 0$ corresponds to staying at the same position all the time.

Table I. Statistics of Our Experiments

Scene	Triangles	Visible Photons Ratio	Time	Mutation Size
Cognac0	16456	103.1	120	0.274
Cognac1	16456	363.8	120	0.176
Cognac2	16456	885.2	120	0.174
Cognac3	16456	910.4	120	0.168
Cognac4	16456	3284.2	120	0.056
Pocket Watch	152434	270.3	90	0.194
Room	160400	41.9	60	0.253
Cornell w/door	730	55.9	90	0.096
Cornell	36	2.0	90	9.852
Box	3462	3.9	90	3.877
Buddha (far)	378731	6.67	90	0.173
Buddha (near)	378731	54.4	90	0.106

The table shows the number of triangles, the ratio of the number of visible photons in total between our method/uniform sampling (larger value means more photons are visible in our method), the rendering time in minutes, and the adaptive mutation size at the end of the rendering process. Cognac(0-4) correspond to different zoom ratios (far to near). Buddha (far/near) corresponds to the far/near viewpoint.

The acceptance probability of a mutated path is easily computed since the mutation is symmetric and $V(\bar{u}) = 1$ in our method. Specifically, given a set of mutations as a vector $\Delta \bar{u} = (\Delta u, \dots, \Delta u)$, the acceptance probability is

$$a(\bar{u} + \Delta \bar{u} \leftarrow \bar{u}) = \frac{F(\bar{u} + \Delta \bar{u})}{F(\bar{u})} = \frac{V(\bar{u} + \Delta \bar{u})}{V(\bar{u})} = V(\bar{u} + \Delta \bar{u}), \quad (10)$$

which simply means that a mutation is accepted if the mutated path is visible. In contrast to existing Markov chain Monte Carlo rendering methods, there is no need for generating another random number to decide whether we accept a mutation or not.

We compute the acceptance ratio, A_i , by counting the number of accepted mutations and dividing this value by the total number of mutations. We then update θ_i as

$$\theta_{i+1} = \theta_i + \gamma_i(A_i - A^*), \quad (11)$$

where $\gamma_i = 1/i$ and $\theta_1 = 1$. The intuition behind this equation is that the acceptance ratio that is too large ($A_i - A^* > 0$) would indicate that the mutation size is too small, thus we increase the mutation size, and likewise the acceptance ratio that is too small ($A_i - A^* < 0$) indicates that the mutation size is too large, so we decrease the mutation size. Note that the difference $A_i - A^*$ can never converge to zero in some scenes, for example, a scene where all paths are visible ($A_i = 100\%$). However, using $\gamma_i = 1/i$ ensures that we always satisfy Eq. (7). Another condition for the convergence of adaptive Markov chain Monte Carlo sampling, *bounded convergence*, requires that the product of the state space and the space of mutations be finite, which is always satisfied in practice since floating point numbers have finite state space. The only parameter, A^* , is “embedded” into the algorithm and users will not touch it, thus our algorithm is parameter free.

6. ALGORITHM

The pseudocode of the algorithm is shown in Figure 3. The main modification to the existing implementation of progressive photon mapping in our case is to change the photon tracing procedure to use a set of given random numbers instead of generating them on-the-fly. The rest of the algorithm was implemented without any large modification to the existing framework since we do not need to compute probability density of paths in our importance function.

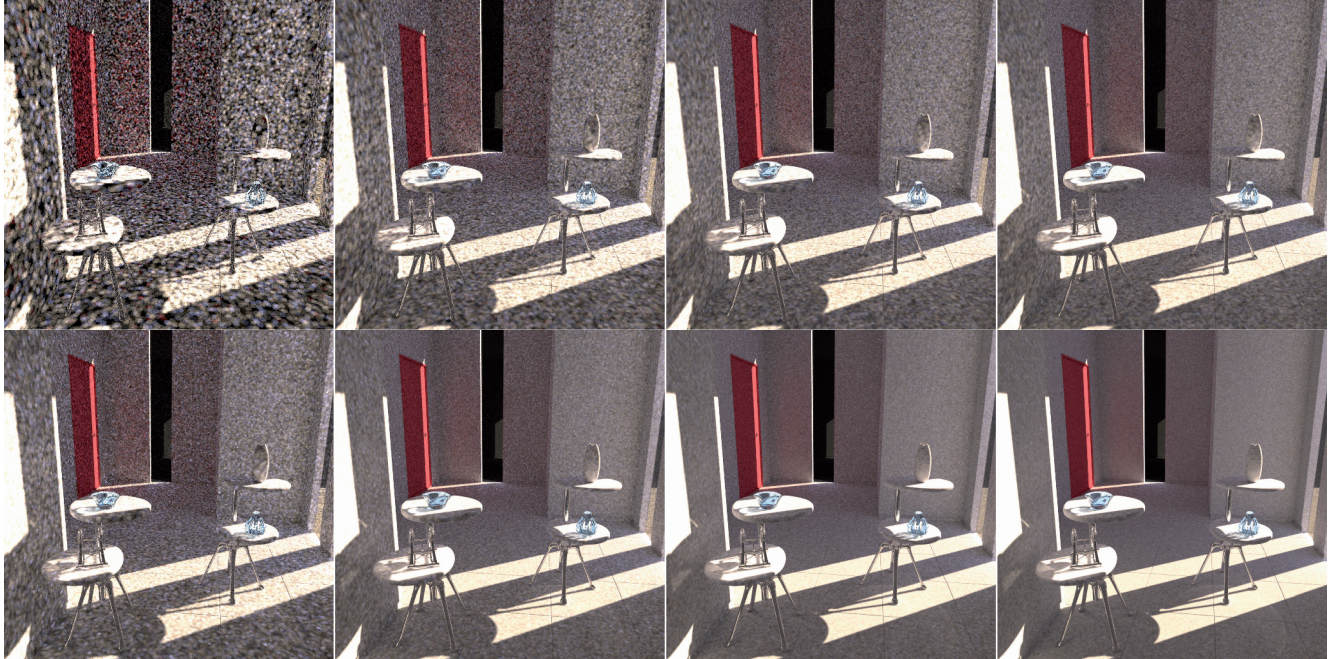


Fig. 5. Sequences of rendered images of a room (with permission of Youichi Kimura) illuminated by a directional light source. The top row shows the results with uniform sampling and the bottom row shows the results with our method using the same rendering time (1, 15, 30, and 60 min from left to right). Our photon tracing method robustly and automatically handles scenes that are considered difficult to render with existing photon tracing approaches. The illumination is coming through the glass window and only photon tracing approaches can handle such paths without ignoring specular reflections and refractions at the window.

7. RESULTS

We have implemented a uniform random photon tracing algorithm and our algorithm using the splatting variation of stochastic progressive photon mapping [Hachisuka and Jensen 2009]. All the scenes have been rendered on a 2.67 GHz Intel Core i7 920 using one core. The alpha value is 0.7 as in the original progressive photon mapping work [Hachisuka et al. 2008]. We trace 200K photons per eye ray tracing pass. The initial radii are manually chosen for each scene as a constant value to get approximately four pixel-wide contribution on the image from each photon at the beginning. This manual tweaking of the initial radii is orthogonal to our claim that the proposed photon tracing algorithm is automatic and parameter free. The resolution of the images is 512^2 except for Figure 13 which uses 640×480 . Table I summarizes various statistics of our experiments. The calculated acceptance ratio is 23.4% for all scenes except the ones where our method does not provide improvement (Cornell and Box).

Figure 1 compares rendered images of a cognac glass illuminated by a directional light source with different zoom ratios. The caustic below the glass exhibits a specular-diffuse-specular path. Due to the fact that this path is generated from a directional light source, existing unbiased Monte Carlo ray tracing methods cannot render this scene since the probability that a path started from the eye hits a directional light source is zero and vice versa. To demonstrate the effect of our sampling method, we zoom into the caustic such that the visible illuminated region becomes increasingly small. This means that many photons land outside the view with uniform photon sampling. Our method focuses photons into the visible region, and we can obtain significantly less noisy images in the same rendering time regardless of the viewpoint. The ratios of visible photons in

Table I also show that our method focuses increasingly more photons compared to uniform photon sampling.

Figure 5 shows a sequence of rendered images of a room illuminated through a glass window by a directional light source. The graph shows relative root mean square errors of the images. The images rendered by our method quickly converge to visually pleasing results compared to uniform photon sampling. Our method also has lower numerical error in the same rendering time as shown in the graph of Figure 6.

Figure 7 highlights the effect of the adaptive Markov chain Monte Carlo method. A mutation size that is too large results in an image with as much noise as the image rendered using uniform sampling, and mutation size that is too small is noisy as well compared to the adaptive mutation size. Our photon tracing method based on adaptive Markov chain Monte Carlo gives us the result shown in the third image from the left of Figure 7 without any tuning of parameters. Note that, in a regular Markov chain Monte Carlo method, we generally cannot know which mutation size works well unless we actually try a wide range of mutation size (0.01 to 4.0 in our example) and compare rendered images. Our method automates this process based on the current scene setting.

We also demonstrate the effect of replica exchange in Figure 11. We rendered the same scene with and without the replica exchange procedure. In this example, we fixed the mutation size to isolate the consequence of using the adaptive mutation. Without replica exchange, photon paths can be trapped within one of the windows for many iterations due to isolated regions of $V(\vec{u}) = 1$. The replica exchange Monte Carlo method alleviates this issue, resulting in more visually plausible images in the same rendering time.

We show the effect of target acceptance ratio in Figure 8. We have rendered the same scene as in Figure 12 using different

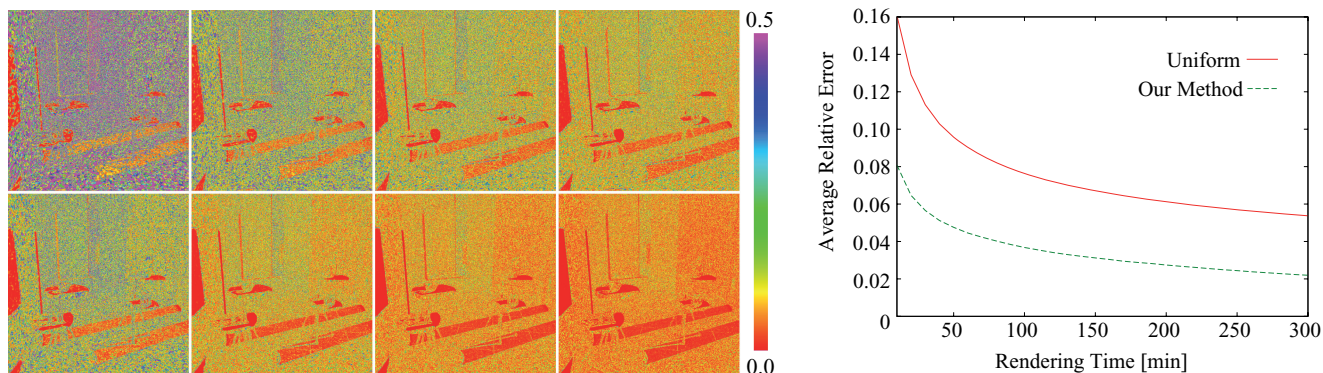


Fig. 6. Color coded error images of Figure 5 and the graph that compares the average errors of rendered images with uniform sampling and our method. Our method not only generates visually smoother images, but also results in more accurate solutions in the same rendering time.



Fig. 7. The effect of adaptive mutation size. A pocket watch is illuminated by a hemispherical light source and a directional light source and is rendered with depth-of-field. Illumination on the dial-plate is due to caustics from the glass cover and the metal lid. The images shown are rendered by uniform random sampling (the leftmost image) and our photon tracing method (right three images) in the same rendering time. The second image uses mutation size that is too small ($d_i = 0.01$) and the fourth image uses mutation size that is too large ($d_i = 4.0$). The adaptive Markov chain Monte Carlo method used in our method (the third image) produces the least noisy result without any parameter tuning.

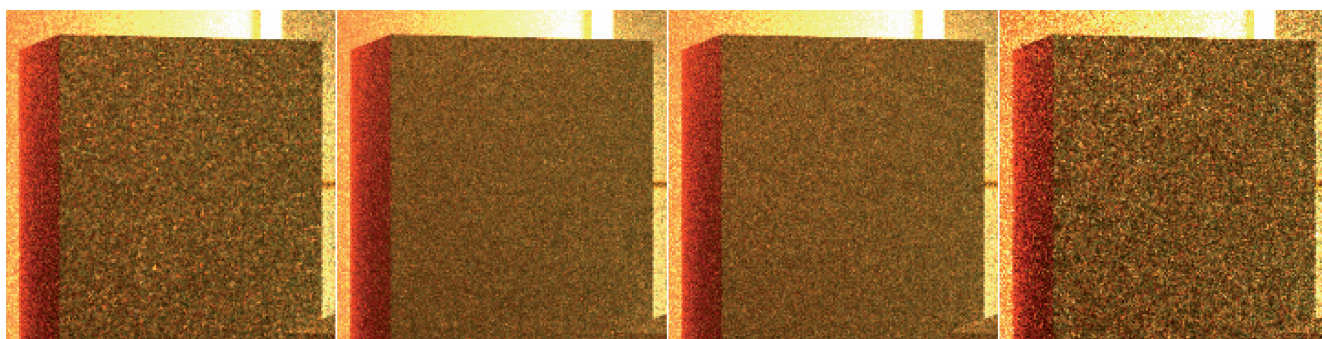


Fig. 8. Close-ups of rendered images using different target acceptance ratios. We have rendered the same scene as in Figure 12 using different target acceptance ratios (from left to right: 5%, 23.4%, 40%, and 90%) using the same rendering time (90 min). Each calculated acceptance ratio achieved is the same as the given target acceptance ratio. The target acceptance ratios closer to 0% or 100% result in slightly noisier images (leftmost and rightmost), but using intermediate values would not affect the efficiency of our algorithm (middle two).

target acceptance ratios (from left to right: 5%, 23.4%, 40%, and 90%). Note that differences in the middle two images (23.4% and 40%) are rather small, which indicates our method is not very sensitive to the target acceptance ratio. However, the images with extreme target acceptance ratios (close to 0% or 100%) are noisier than the image with the target acceptance ratio of 23.4%. As

mentioned earlier, similar results have been observed in other applications in computational statistics [Rosenthal et al. 2008] and we confirmed that this is also the case in our method through numerical examples.

Figure 9 shows examples where our method will not provide a benefit as most of the paths are already visible. For such scenes,

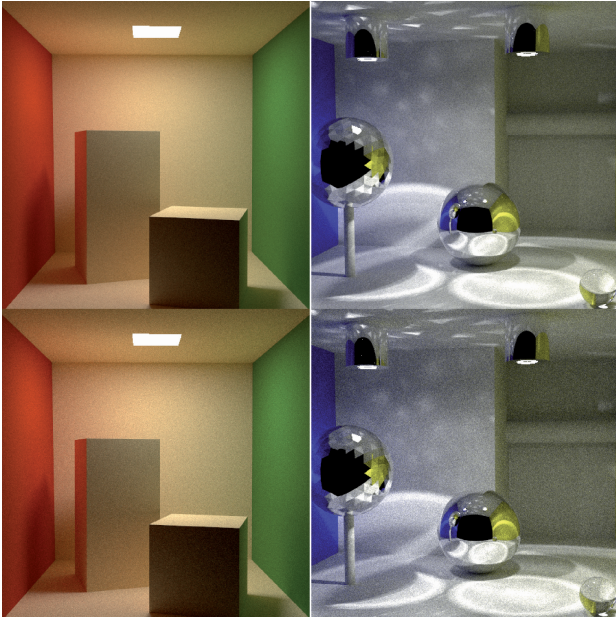


Fig. 9. Rendered images of scenes where our method does not provide benefits. Our method (top) still performs as well as uniform random sampling (bottom) does in the same rendering time (90 min). Note that box scene is the one that is demonstrated to be rendered robustly only with progressive photon mapping [Hachisuka et al. 2008].

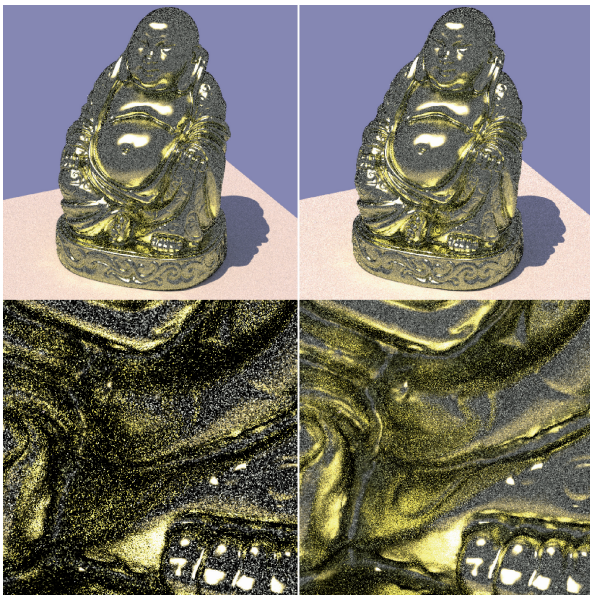


Fig. 10. Example scene where flux variations due to BRDFs are the main source of rendering error. The statue (with permission of VC-ISTI) is rendered using a modified-Phong model with the exponent 100 under a hemispherical light source and a directional light source using uniform random sampling (left column) and our method (right column). The rendering time is 90 min. Although our method shows improvement in the close-up images (bottom row), our method performs approximately the same with the distant viewpoint (top row) as it does not resolve flux variations due to BRDFs.

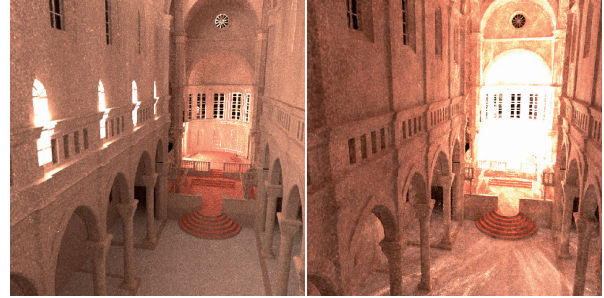


Fig. 11. Effect of replica exchange. We rendered Sibenik cathedral (with permission of Marko Dabrovic) with a directional light source as the only light source in 90 min with (left) and without (right) replica exchange. Without replica exchange, samples can get stuck within a small region (i.e., single window) for long time, resulting in a solution with very high correlation of samples.

sample correlation introduced by a Markov chain Monte Carlo sampler would just result in additional rendering artifacts. The results, however, show no visible negative effect when we compared our method to uniform sampling. This is because the exchange by uniform random sampling happens often in this type of scene, and our algorithm automatically uses uniform random sampling for most of the photons (refer to the pseudocode in Figure 3). At the same time, the adaptive Markov chain Monte Carlo method automatically increases the mutation size to decrease sample correlation. The mutation sizes shown in Table I for Cornell and Box therefore are thus noticeably larger than other scenes.

Table I provides a rough idea of the cases where we will see benefits using our photon tracing method. The column of “Visible Photon Ratio” shows how many times more photons become visible using our method. For the scenes that show no benefit (Cornell and Box), the ratio is less than 10. As we can see in Figure 5 and Figure 6, the room scene already shows some benefit with our method, and the ratio is 41.9, thus the ratio may need to be more than a few tens to obtain improvement using our method.

Since we use a hypercube of random numbers as the sampling space, our method can handle local lighting similar to Metropolis light transport without any modification as shown in Figure 12. Figure 13 visually verifies that our method preserves robustness of progressive photon mapping, thus being able to handle scenes that are difficult for the unbiased methods. None of the images using the unbiased methods is visually converged, thus the differences in the shape of caustics developing in the result of Metropolis light transport and our results are due to error of the solution.

8. DISCUSSION

8.1 Comparisons with Other Importance Functions

In a regular implementation of Metropolis light transport, the importance function is usually given as the brightness of each path [Veach and Guibas 1997; Kelemen et al. 2002]. Using this importance function, each importance sampled path contributes the same brightness to the image. This importance function achieves sampling according to outgoing radiance within each pixel toward the eye. This choice is motivated by picking a probability density function that is proportional to the integrand which results in zero variance.

The important fact is that the samples are distributed over the image. This essentially means that we solve multiple integrations with different integrands at the same time by distributing each

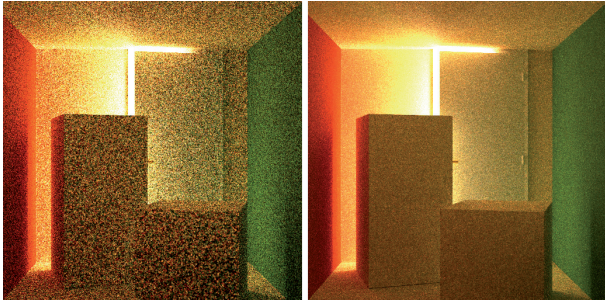


Fig. 12. Cornell box with a small gap with a door. The images are rendered with uniform sampling (left) and our method (right) in 90 min. Our method works under local lighting without any modifications to the algorithm.

sample according the brightness of the path. As a result, bright pixels get more samples compared to dark pixels as we mentioned in Section 3.2, which results in poor stratification of samples over the image. Energy redistribution path tracing [Cline et al. 2005] improves stratification of samples by starting multiple independent chains from stratified screen-space samples on the image. However, even if those initial points are well distributed, succeeding Markov chains can get stuck in paths with very large brightness for many iterations. Cline et al. thus proposed postprocess filtering, which unfortunately makes the algorithm inconsistent (i.e., does not converge to the correct solution). Although our target function does not provide stratification on the image, the function does not have any local peak that is prone to this issue. Isolated visible paths can still lead to this issue, however, the replica exchange Monte Carlo method alleviates the problem in such cases since the exchange results in a completely different path as soon as we find another visible path by uniform sampling.

Interesting observations on alternative target functions have recently been made by Hoberock and Hart [2010]. They proposed a multipass algorithm that adjusts the importance function using information from previous passes, such as brightness and variance of each pixel. Their work also supports our claim that just using the brightness of each path is not necessarily the optimal choice. They additionally pointed that a constant importance function does not work because no importance sampling will be employed, which may be confusingly similar to our importance function. The important distinctions are that our function is applied to photon tracing and our function returns 0 for invisible photon paths. Since photons are naturally distributed according to incoming radiance if proper local importance sampling and Russian Roulette are done (i.e., photon density is equal to radiance), our method still employs importance sampling of incoming radiance, even with our simple target function.

8.2 Limitations

One limitation of our method is that it ignores flux variations due to BRDFs (Figure 9). For example, if a scene has a highly glossy material, just using visibility information will not resolve this flux variation due to the glossy BRDF lobe. Figure 10 demonstrates such an example scene, where most of noise is due to a highly glossy BRDF. Although our method still provides visible improvement when we render the close-up of the same scene, it does not resolve noise due to the glossy BRDF. Note that this particular example scene itself might be efficiently rendered with other methods, such as path tracing with the next event estimation, but we chose this scene to highlight the limitation of our method. One possible solution is to

use stochastic progressive photon mapping to perform importance sampling of a BRDF from the eye, which resolves this flux variation. However, this solution is not perfect. Ideally, we would like to sample photon paths according to its contribution to the image, which cannot be achieved by stochastic progressive photon mapping in a general setting.

Another limitation is that the adaptive procedure is done globally. Using locally adaptive mutation parameters might improve convergence speed as was proposed in existing computational statistics literatures [Andrieu and Thoms 2008]. For example, we might be able to use an adaptive grid to store mutation parameters locally, and use and update these parameters according to the current state (i.e., position in the hypercube). This, however, could be challenging as our sampling space is in high-dimensional space, where the cost of storing any local estimation including adaptive mutation parameters is often prohibitive and reliable estimation is difficult due to the curse of dimensionality.

As we mentioned earlier, our algorithm does share the limitation with the other Markov Monte Carlo chain rendering algorithms that the samples are not stratified over an image. It is interesting to investigate as future work whether the adaptive importance function proposed by Hoberock and Hart [2010] is applicable to our method in order to improve stratification.

8.3 Dynamic Target Distribution

One theoretical difficulty of applying any Markov chain Monte Carlo method to progressive photon mapping is that the target distribution, thereby the importance function, changes as the number of samples increases. This is because progressive photon mapping updates the radii of the measurement points to ensure convergence to the correct solution. In our approach, this results in changes of the region where $V(\vec{u}) = 1$. Although we have not found any apparent failure cases, the theoretical behavior of Markov chain Monte Carlo methods on the dynamic target distribution in progressive photon mapping is not fully analyzed. Our combination with the adaptive Markov chain Monte Carlo method further complicates this theoretical validation. Convergence of the normalization term may also require careful theoretical analysis. In this article, we thus do not claim provable convergence to the correct solution using our photon tracing algorithm.

However, since we always use the current radii to distribute the photon power, the contribution is at least computed based on the current distribution. It is only the stationary sample distribution that is not analyzed. This separation might be helpful for further theoretical analysis. We also believe that, even without a formal theoretical validation, our method will be useful for many practical applications that do not require theoretical guarantees of consistency. In the end, provable convergence is only a theoretically appealing property as we cannot take infinite number of samples in practice. We see practical benefits of using our method through numerical experiments as we have demonstrated.

9. CONCLUSION

We have presented a new photon tracing algorithm using a simple and effective importance function based on the visibility of photon paths. Our algorithm uses recent developments in Markov chain Monte Carlo methods. The resulting algorithm does not have any parameters that require fine tuning by the user, and its implementation is strikingly simple. We have demonstrated that our algorithm efficiently handles scenes that are difficult for existing photon tracing approaches, while still keeping the efficiency for simple

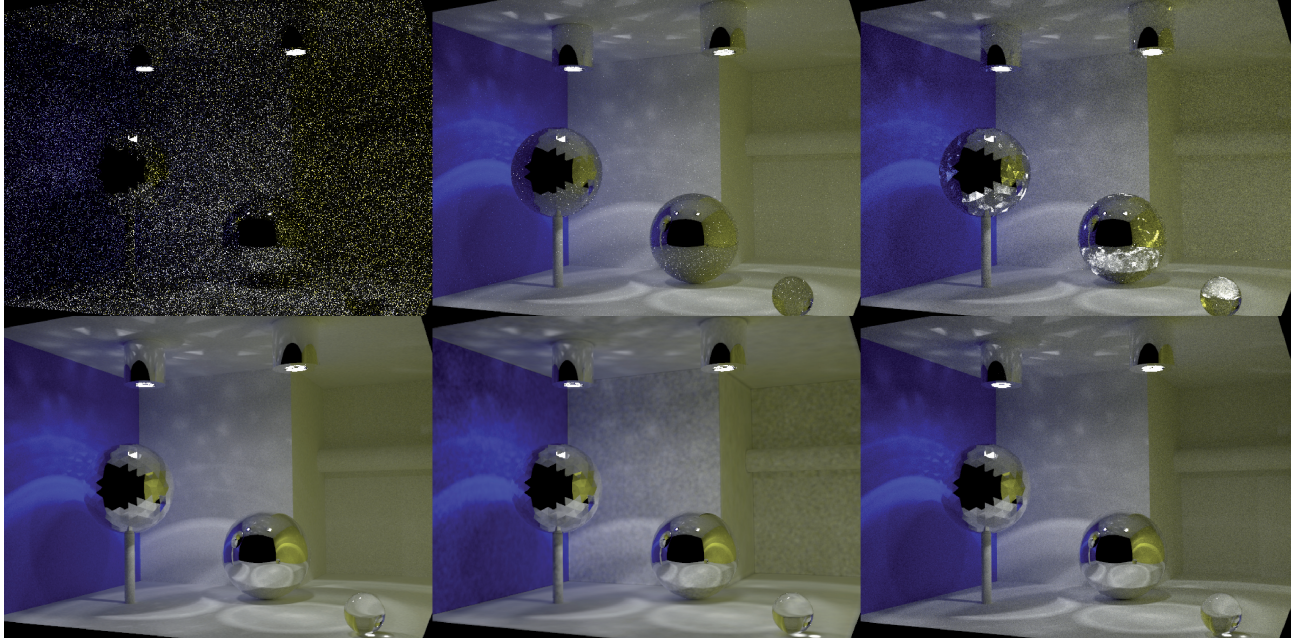


Fig. 13. Equal-time comparisons with other rendering methods (four hours, except for conventional photon mapping for which rendering time is limited by memory consumption). The images are rendered with, from left to right, top row: path tracing, bidirectional path tracing, Metropolis light transport, and bottom row: original progressive photon mapping, conventional photon mapping with 20M photons, and progressive photon mapping with our photon tracing method. Our method does not impair the robustness of original progressive photon mapping. The small differences in the result of our method are due to the use of different rendering systems.

scenes. The combination of our algorithm and progressive photon mapping is an effective, unified, and robust solution to many light transport configurations. Although we used progressive photon mapping in this article, we expect that the same importance function can be used for the original photon mapping and other Monte Carlo ray tracing methods to improve the efficiency of photon tracing.

ACKNOWLEDGMENTS

We would like to thank Youichi Kimura (Studio Azurite) for providing us the room model. The Buddha statue model is courtesy of the Visual Computing Laboratory at the Instituto di Scienza e Tecnologie dell'Informazione (VC-ISTI) and the Sibenik model is courtesy of Marko Dabrovic.

REFERENCES

- ANDRIEU, C. AND ROBERT, C. P. 2001. Controlled mcmc for optimal sampling. Tech. rep. 0125, Cahiers de Mathématiques du Ceremade, UniversitéParis-Dauphine.
- ANDRIEU, C. AND THOMS, J. 2008. A tutorial on adaptive MCMC. *Statist. Comput.* 18, 4, 343–373.
- ARVO, J. 1986. Backward ray tracing. In *ACM SIGGRAPH Course Notes, Developments in Ray Tracing*, 259–263.
- CLINE, D., TALBOT, J., AND EGBERT, P. 2005. Energy redistribution path tracing. *ACM Trans. Graph.* 24, 3, 1186–1195.
- DUTRÉ, P., LAFORTUNE, E., AND WILLEMS, Y. 1993. Monte carlo light tracing with direct computation of pixel intensities. In *Proceedings of Compugraphics Conference*. 128–137.
- FAN, S., CHENNEY, S., AND CHI LAI, Y. 2005. Metropolis photon sampling with optional user guidance. In *Proceedings of the 16th Eurographics Symposium on Rendering Techniques*. Eurographics Association, 127–138.
- HAARIO, H., SAKSMAN, E., AND TAMMINEN, J. 2001. An adaptive metropolis algorithm. *Bernoulli* 7, 2, 223–242.
- HACHISUKA, T. AND JENSEN, H. W. 2009. Stochastic progressive photon mapping. In *ACM SIGGRAPH Asia Papers*. ACM, New York, 1–8.
- HACHISUKA, T., OGAKI, S., AND JENSEN, H. W. 2008. Progressive photon mapping. *ACM Trans. Graph.* 27, 5, Article 130.
- HOBEROCK, J. AND HART, J. C. 2010. Arbitrary importance functions for metropolis light transport. *Comput. Graph. Forum* 29, 6, 1993–2003.
- IBA, Y. 2001. Extended ensemble monte carlo. *Int. J. Modern Phys. C* 12, 05, 623–656.
- JENSEN, H. W. 1996. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques*. Springer, 21–30.
- KAJIYA, J. T. 1986. The rendering equation. *ACM Comput. Graph.* 20, 4, 143–150.
- KELEMEN, C., SZIRMAY-KALOS, L., ANTAL, G., AND CSONKA, F. 2002. A simple and robust mutation strategy for the metropolis light transport algorithm. *Comput. Graph. Forum*. 531–540.
- KITAOKA, S., KITAMURA, Y., AND KISHINO, F. 2009. Replica exchange light transport. *Comput. Graph. Forum* 28, 8, 2330–2342.
- LAFORTUNE, E. P. AND WILLEMS, Y. D. 1993. Bi-Directional path tracing. In *Proceedings of Compugraphics Conference*. H. P. Santo, Ed., 145–153.
- ROBERTS, G. O., GELMAN, A., AND GILKS, W. R. 1997. Weak convergence and optimal scaling of random walk metropolis algorithms. *Ann. Appl. Probab.* 7, 1, 110–120.
- ROSENTHAL, J. S., BROOKS, S., GELMAN, A., JONES, G., AND L. MENG, X. 2008. Optimal proposal distributions and adaptive MCMC. In *MCMC Handbook*.

- SEGOVIA, B., IEHL, J.-C., AND PEROCHE, B. 2007. Coherent metropolis light transport with multiple-try mutations. Tech. rep. RR-LIRIS-2007-015.
- SWENDSEN, R. H. AND WANG, J.-S. 1986. Replica monte carlo simulation of spin-glasses. *Phys. Rev. Lett.* 57, 21, 2607+.
- VEACH, E. 1998. Robust monte carlo methods for light transport simulation. Ph.D. thesis, Stanford, CA. J. Guibas.
- VEACH, E. AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for monte carlo rendering. *ACM Comput. Graph.* 419–428.
- VEACH, E. AND GUIBAS, L. J. 1997. Metropolis light transport. *ACM Comput. Graph.* 65–76.

Received October 2010; revised March 2011; accepted June 2011