

# Improved Stochastic Progressive Photon Mapping with Metropolis Sampling

Jiating Chen<sup>1,2,3,4</sup>, Bin Wang<sup>1,3,4</sup> and Jun-Hai Yong<sup>1,3,4</sup>

<sup>1</sup>School of Software, Tsinghua University, Beijing, China

<sup>2</sup>Department of Computer Science and Technology, Tsinghua University, Beijing, China

<sup>3</sup>Key Laboratory for Information System Security, Ministry of Education of China, Beijing, China

<sup>4</sup>Tsinghua National Laboratory for Information Science and Technology, Beijing, China

---

## Abstract

*This paper presents an improvement to the stochastic progressive photon mapping (SPPM), a method for robustly simulating complex global illumination with distributed ray tracing effects. Normally, similar to photon mapping and other particle tracing algorithms, SPPM would become inefficient when the photons are poorly distributed. An inordinate amount of photons are required to reduce the error caused by noise and bias to acceptable levels. In order to optimize the distribution of photons, we propose an extension of SPPM with a Metropolis-Hastings algorithm, effectively exploiting local coherence among the light paths that contribute to the rendered image. A well-designed scalar contribution function is introduced as our Metropolis sampling strategy, targeting at specific parts of image areas with large error to improve the efficiency of the radiance estimator. Experimental results demonstrate that the new Metropolis sampling based approach maintains the robustness of the standard SPPM method, while significantly improving the rendering efficiency for a wide range of scenes with complex lighting.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation  
I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism

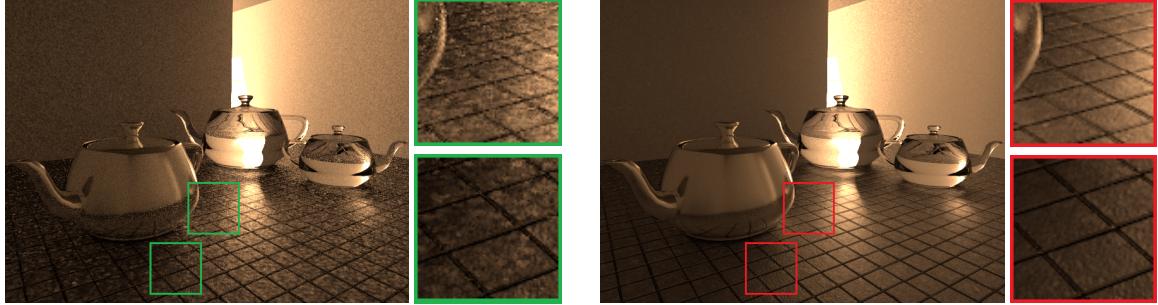
---

## 1. Introduction

Efficient computation of global illumination is important for rendering realistic synthetic scenes in computer graphics. Recent years have witnessed significant progress in solving the rendering equation introduced by Kajiya [Kaj86] for different types of lighting [DBBS06]. Unbiased Monte Carlo methods have become a popular solution to the rendering equation without any approximation. However, they cannot efficiently handle some complex illumination settings including specular-diffuse-specular (SDS) paths. For example, handling specular reflection/refraction of caustics from small light sources is particularly difficult for the unbiased methods, as analyzed in Hachisuka *et al.* [HOJ08].

Since SDS paths are common in light transport within realistic optical systems and natural scenes, it is greatly desirable to efficiently compute such illumination for industrial purposes, such as lamp manufacture and building design. Progressive photon mapping (PPM) [HOJ08] is considered

the first biased technique to robustly solve this issue with a bounded memory consumption. Compared with the normal photon mapping [Jen96], PPM is capable of computing the correct radiance (e.g., detailed caustics) without having to store the photons. Stochastic progressive photon mapping (SPPM) [HJ09] extends PPM to simulate global illumination with distributed ray tracing effects [CPC84], while maintaining the robustness of PPM. However, both PPM and SPPM approaches would become highly inefficient when the photons are poorly distributed, especially when only a small part of the photons can arrive at visible regions. This situation is not uncommon in practice. Normally, photons are emitted randomly from light sources and deposited as they interact with scene surfaces, but most of them may be invisible from the viewpoint (Figure 1). In addition, the visible photons are unevenly distributed due to occlusion. The photon density in some visible regions tends to be relatively low (Figure 2). Without sufficient photons, the radiance error of these image areas is inefficient to reduce to acceptable levels.



**Figure 1:** Room with difficult visibility issues. Using uniform random sampling of photons, the SPPM method performs poorly since only 5.0% of the light paths can arrive at visible regions. Thanks to local exploration, the percentage of the visible light paths increases to 53.3% using Metropolis sampling of photons. Therefore, our Metropolis sampling approach (right) used in SPPM method produces less noise compared with the uniform random sampling (left) in the same rendering time.

In this paper, we present a simple extension of SPPM with a Metropolis-Hastings approach [MRR<sup>\*</sup>53, Has70] to optimize the distribution of photons. The main advantage of the Metropolis sampling is that the visible light paths that contribute to the results can be locally explored by mutating the current visible light path. In our Metropolis sampling strategy, we introduce a well-designed scalar contribution function, targeting at specific parts of the image areas with large error. As expected, experimental results show that the percentage of the visible light paths that contribute to the result dramatically increases for a wide range of scenes with complex lighting. Our sampling method can also adaptively trace more photons to the problematic areas with low photon density. Therefore, our approach significantly improves the rendering efficiency compared with the standard SPPM method using uniform random sampling of photons.

## 2. Related Work

**Unbiased Ray Tracing** The unbiased ray tracing methods, such as path tracing [Kaj86] and bi-directional path tracing [LW93, VG95], are the general-purpose solutions to the rendering equation using a large number of Monte Carlo samples. In order to reduce the number of samples, several adaptive rendering methods [HJW<sup>\*</sup>08, ODR09, SSD<sup>\*</sup>09, ETH<sup>\*</sup>09, CWW<sup>\*</sup>11] have been developed to efficiently simulate the distributed ray tracing effects. Unfortunately, these methods cannot robustly simulate SDS paths.

The reuse of light paths is another adaptive alternative of exploiting useful light paths to render complex illumination effects. Metropolis Light Transport (MLT) [VG97] adopts Metropolis sampling as the sampler for path tracing. Once high contribution paths are found, nearby paths with high contribution will be likely to be explored as well. Then several algorithms [CTE05, LFCC07, KKK09] were proposed to enhance the MLT approach. Nevertheless, all these methods are still inefficient in rendering SDS paths from small light sources (e.g., point lights and directional lights).

The original MLT approach has been extended in different ways. The start-up bias problem was analyzed in [SKDP99]. Kelemen *et al.* [KSKAC02] proposed a simplification of MLT which increases the acceptance rate by using a user-defined mutation strategy. Hoberock and Hart [HH10] showed that the scalar contribution function in the MLT method can be similarly programmable. In our method, we introduce a well-designed scalar contribution function as our Metropolis sampling strategy, which is demonstrated to be compatible with the standard SPPM approach [HJ09].

**Photon Mapping** Photon mapping [Jen96] is one of the popular particle tracing algorithms, which involves a photon scattering pass and a final gathering pass to efficiently simulate global illumination. This method robustly handles SDS paths, since it can loosely connect SDS paths by means of photon density estimation. However, photon mapping would suffer from insufficient photons due to bounded memory, resulting in the blurring of sharp features. To address this issue, progressive photon mapping (PPM) [HOJ08] was presented as a progressive refinement extension without storing the photons, making it possible to converge to the correct solution. Stochastic progressive photon mapping (SPPM) [HJ09] computes the average radiance over a region for robustly rendering distributed ray tracing effects. However, both PPM and SPPM approaches tend to be inefficient since the photons are likely to be poorly distributed. To improve photon tracing, Metropolis photon sampling [FCL05] was introduced to target at important light paths, producing better photon distribution. Unfortunately, this method fails to solve SDS paths from small light sources, because it uses sampling on the exact path space to specify the important light paths. A recent approach that is most related to our work is the adaptive photon tracing algorithm [HJ11], which uses photon path visibility as the importance function. In contrast, our scalar contribution function further considers the uneven distribution of the visible photons. Therefore, our method is more efficient to handle this kind of illumination settings.

### 3. Overview

#### 3.1. Stochastic progressive photon mapping

Stochastic progressive photon mapping (SPPM) [HJ09] is a multi-iteration method that converges to the correct radiance by accumulating statistics of photons over a region. There are two passes in each iteration. The first distributed ray tracing pass traces rays from camera and updates all the non-specular hit points for each region. The following photon tracing pass traces photons from the light sources and then updates statistics over the regions. In the  $i$ -th iteration, the radiance estimator approximates the average radiance value  $L(S)$  over the region  $S$  (e.g., a pixel footprint) as:

$$L(S) \approx \frac{\tau_i(S)}{N_e(i)\pi R_i(S)^2}, \quad (1)$$

where  $\tau_i(S)$  is the accumulated flux times BRDF over the region  $S$ ,  $N_e(i)$  is the number of emitted photons after  $i$  iterations, and  $R_i(S)$  is the shared search radius. The shared statistics over the region  $S$  is updated as:

$$N_{i+1}(S) = N_i(S) + \alpha M_i(\vec{x}_i) \quad (2)$$

$$R_{i+1}(S) = R_i(S) \sqrt{\frac{N_i(S) + \alpha M_i(\vec{x}_i)}{N_i(S) + M_i(\vec{x}_i)}} \quad (3)$$

$$\phi_i(\vec{x}_i, \vec{w}_i) = \sum_{p=1}^{M_i(\vec{x}_i)} f_r(\vec{x}_i, \vec{w}_i, \vec{w}_p) \phi_p(\vec{x}_p, \vec{w}_p) \quad (4)$$

$$\tau_{i+1}(S) = (\tau_i(S) + \phi_i(\vec{x}_i, \vec{w}_i)) \frac{R_{i+1}(S)^2}{R_i(S)^2}, \quad (5)$$

where  $N_i(S)$  is the accumulated photon count within the search radius,  $\alpha \in (0, 1)$  is a user-defined parameter ( $\alpha = 0.8$  in this paper),  $\vec{x}_i$  is a hit point that is generated within the region  $S$  by distributed ray tracing,  $M_i(\vec{x}_i)$  is the photon count within the radius during the iteration  $i$ ,  $\vec{w}_i$  is the outgoing direction from the hit point  $\vec{x}_i$ ,  $f_r$  is the BRDF,  $\phi_p(\vec{x}_p, \vec{w}_p)$  is the flux of photon  $p$ ,  $\vec{w}_p$  is the incoming direction of photon  $p$ , and  $\phi_i(\vec{x}_i, \vec{w}_i)$  is the accumulated flux times BRDF during the iteration  $i$ . More specific issues are discussed in [HJ09].

#### 3.2. SPPM with Metropolis sampling

Our method rarely requires any modification to the updating procedure of the statistics, but adopts a Metropolis sampling to optimize the photon distribution. During the photon tracing pass, each path  $X$  is a sequence of points  $x_0x_1\dots x_k$ , where  $x_0$  is on the light source, the rest are on the scene surfaces, and  $k \geq 1$  is the path length. For each path we design a novel scalar contribution function  $I(X)$ , and Metropolis sampling generates path  $X_i$  with probability proportional to  $I(X_i)$ :

$$P(X_i) = \frac{I(X_i)}{b}, \quad (6)$$

where  $b = \int_{\Omega} I(X)d\Omega$ . In practice, we generate a sequence of paths  $X_0, X_1, \dots, X_N$ , where each  $X_i$  is obtained by a mutation to the preceding path  $X_{i-1}$ . The acceptance probability of the mutation depends on the scalar contribution functions of the new and old paths. As the path  $X_i$  is sampled, the flux of each photon along this path needs to be normalized as:

$$\phi'_p(\vec{x}_p, \vec{w}_p) = \frac{\phi_p(\vec{x}_p, \vec{w}_p)}{P(X_i)} = \phi_p(\vec{x}_p, \vec{w}_p) \frac{b}{I(X_i)}. \quad (7)$$

Finally, the flux is accumulated to the corresponding regions using Equation (4) with the *expected values* enhancement [VG97]. We summarize this method in Algorithm 1.

---

#### Algorithm 1: Metropolis sampling for SPPM

---

```

1 Estimate the normalized constant  $b = \int_{\Omega} I(X)d\Omega$ ;
2  $X_0 \leftarrow \text{InitialSample}();$ 
3  $I(X_0) = \text{PhotonTracing}(X_0);$ 
4 for  $i = 1$  to  $N$  do
5    $X' \leftarrow \text{Mutate}(X_{i-1});$ 
6    $I(X') = \text{PhotonTracing}(X');$ 
7    $a = \text{AcceptProbability}(X' | X_{i-1});$ 
8   if  $\text{Random}() < a$  then
9      $X_i = X'; I(X_i) = I(X');$ 
10   end
11   else
12      $X_i = X_{i-1}; I(X_i) = I(X_{i-1});$ 
13   end
14    $\text{RecordSample}(X_{i-1}, (1-a)b/I(X_{i-1}));$ 
15    $\text{RecordSample}(X', ab/I(X'));$ 
16 end

```

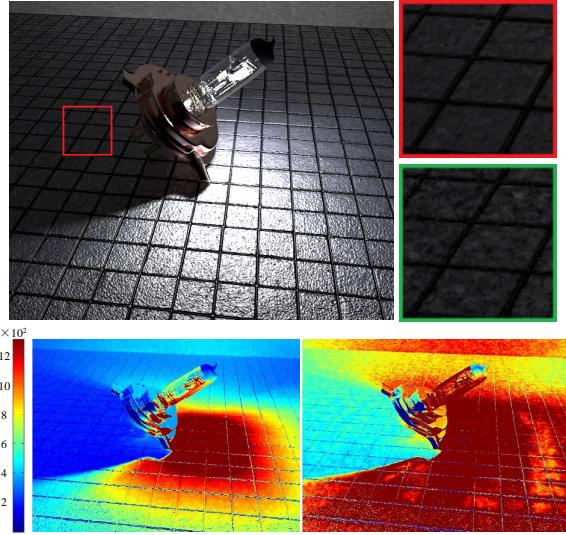
---

## 4. Metropolis Sampling for SPPM

### 4.1. Photon distribution and rendering efficiency

Using uniform random sampling, the SPPM method samples light paths without considering the viewpoint and scene features, hence easily resulting in poor distribution of photons. One problematic case is that most of the light paths cannot deposit photons in the visible regions (Figure 1), leading to too much unnecessary computation time for tracing invisible photons. This is common in scene configurations such as illumination coming through a small gap. Another case is the uneven distribution of visible photons due to occlusion, causing the result error of some image areas to be large. Hachisuka *et al.* [HJJ10] presented an error estimation framework for PPM. However, error estimation for SPPM remains uninvestigated as far as we know. Based on the experiments, we can observe that there are usually two kinds of image areas where the error tends to be large. One are the areas with distributed ray tracing effects, and the other one are the areas with low photon density (Figure 2). We do

not consider the former one since SPPM can robustly handle the distributed ray tracing effects. For the latter one, in order to reduce the error due to noise using limited photons, it requires to increase the search radius for photon lookups. But this would increase the error due to bias in return, blurring illumination details. In this case, the SPPM method also becomes inefficient. In order to improve the rendering efficiency, our adaptive method attempts to optimize the photon distribution considering the above two cases simultaneously.



**Figure 2:** Bulb scene with uneven photon distribution. Using uniform random sampling, the photon density in the red highlighted areas is relatively low (left in the second row). While the photon density in these areas significantly increases since our Metropolis sampling method adaptively traces more photons there (right in the second row). Therefore, our adaptive photon tracing method (the red close-up) is more efficient to reduce the error in these areas compared with uniform random sampling (the green close-up).

#### 4.2. Scalar contribution function

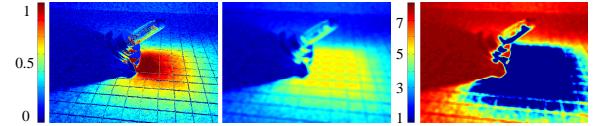
Metropolis sampling remains unbiased for any scalar contribution function  $I(X)$ , but a poor choice of  $I(X)$  would reduce the rendering efficiency. The original MLT algorithm [VG97] samples light paths with probability proportional to brightness. Hoberock and Hart [HH10] showed that this luminance-based function is undesirable since it would lead to a poor estimation for common scenes. Based on the analysis in Section 4.1, we design the scalar contribution function based on the photon distribution of the scenes.

We generally start by coarsely tracing light paths using uniform random sampling and estimating a photon density map (left in Figure 3). Then we remove the extreme low and high values by clamping each value  $v$  between  $v_{low}$  and  $v_{high}$  (e.g., 5% and 60% of the range, respectively), and further

remove the high spatial frequencies using a low-pass filter (middle in Figure 3). Finally, we transform it to a scalar importance map (right in Figure 3):

$$s = \begin{cases} 1 & \text{if } v \geq v_{mid}, \\ 1 + \beta(e^{1-v/v_{mid}} - 1) & \text{otherwise,} \end{cases} \quad (8)$$

where  $v_{mid} = (v_{low} + v_{high})/2$ , and  $\beta > 0$  is a user-supplied parameter. Intuitively speaking, this equation reads as follows. The scalar variable  $s$  remains the same when the corresponding photon density tends to be sufficient ( $v \geq v_{mid}$ ). If the photon density would be insufficient ( $v < v_{mid}$ ), the variable  $s$  increases as the photon density  $v$  becomes lower.



**Figure 3:** The photon density map (left) is clamped and filtered to a smoothed map (middle), and then transformed to a scalar importance map (right). Based on the scalar importance map, Metropolis sampling can trace more photons to the areas with low photon density, as shown in Figure 2.

We use the scalar importance map to design the scalar contribution function  $I(X)$  for each path  $X = x_0x_1\dots x_k$ :

$$I(X) = \begin{cases} 0 & \text{if } k = 0, \\ \max_{i=1}^k \{V(x_i)s(x_i)\} & \text{if } k > 0, \end{cases} \quad (9)$$

where  $V(x_i)$  is the visibility function of point  $x_i$ :  $V(x_i) = 1$  if  $x_i$  is visible from the viewpoint, and  $V(x_i) = 0$  otherwise. When  $x_i$  is visible,  $s(x_i)$  is the pixel value corresponding to  $x_i$  in the scalar importance map. Thus,  $I(X)$  is determined by the visible point that is on path  $X$  and with the lowest photon density. This is desirable. When none of these points is visible,  $I(X) = 0$  and Metropolis sampling would refuse to sample its neighboring light paths. In this way, the ratio of the visible light paths can be improved. When some of the points are at the visible locations where the photon density is low,  $I(X)$  is accordingly large and the path  $X$  can be locally explored by Metropolis sampling, tracing more photons to the local areas with low photon density (see Figure 2). However, it should be noted that increasing  $I(X)$  for some areas would be at the cost of increasing error in other areas. Therefore, we cannot increase  $I(X)$  for the problematic areas without limit. In practice, we set  $\beta = 10$  in Equation (8) to achieve a good trade-off for the examples in this paper.

The scalar contribution function can be further tailored to some user-specified light paths  $X_s$  to enhance specific light transport effects without introducing additional bias:

$$I'(X_s) = \lambda I(X_s), \quad (10)$$

where  $\lambda > 1$  is a user-supplied parameter. For example, we use it to emphasize the caustic effects in this paper with  $\lambda =$

4 when the light paths  $X_s$  generate visible caustic photons after they interact with specular surfaces.

### 4.3. Implementation details

**Initialization:** In this step, we need to compute the integral of the scalar contribution function  $b = \int_{\Omega} I(X) d\Omega$ . We first use 4 iterations of SPPM to compute the scalar importance map using Equation (8). Then we use 4 more iterations to estimate  $b \approx \frac{1}{n} \sum_{i=1}^n I(X_i)$ , where  $n$  is the path count, and  $I(X_i)$  for each path  $X_i$  can be computed using Equations (9) and (10). After that, we start the Metropolis sampling process.

**Start-up bias:** The initial path  $X_0$  is required to be sampled with probability proportional to  $I(X)$ . Otherwise, it would lead to the start-up bias. We adopt the initial sample selecting strategy of the MLT implementation in PBRT [PH10] to avoid this problem. We first generate a uniform random number between 0 and the contribution sum computed in the initialization step, and then loop over the computed light paths again until the one whose contribution causes the accumulated contribution sum to pass the value is found. This sample can be an appropriate initial path  $X_0$ .

**Mutation strategy:** We use the mutation rule described by Kelemen *et al.* [KSKAC02] for our mutations in the path space. This mutation rule has two kinds of mutations, global mutation and local mutation, and a user-supplied parameter  $\epsilon$  as the probability of carrying out the global mutation. The global mutation is to discard the current light path and uniformly generate a new one at random. The local mutation is to perturb the current path  $X = x_0x_1x_2\dots x_k$  as:

$$x'_i = x_i \pm s_2 e^{-\log(s_2/s_1)\xi}, \quad (11)$$

where  $\xi$  is a uniform random number in  $[0, 1]$  and the samples are expected in  $[s_1, s_2]$ . In the experiments, we set  $\epsilon = 0.1$ ,  $s_2 = 16s_1$ ,  $s_2 = 1/1024$  for point  $x_0$  on the light source, and  $s_2 = 1/64$  for the rest points. We set a smaller mutation for  $x_0$  since it could be easily perturbed for the area lights.

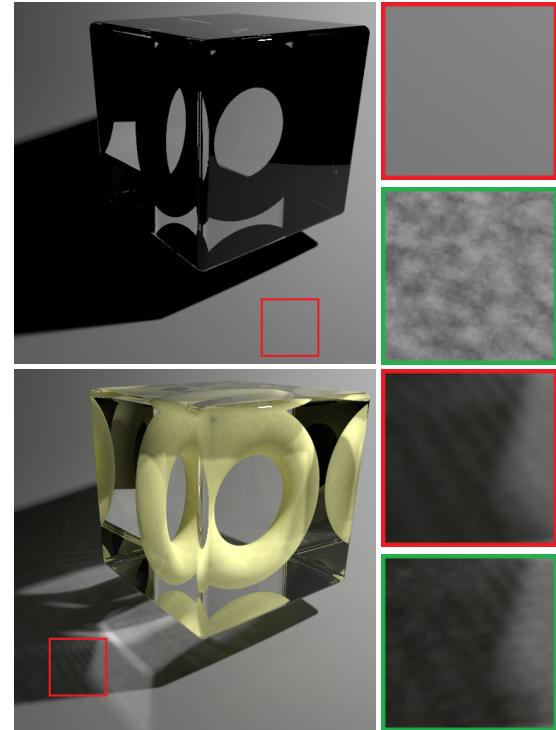
**Acceptance probability:** Since the above mutations are symmetric, the transition probability density is the same in both directions, i.e.,  $T(X'|X_{i-1}) = T(X_{i-1}|X')$ . Therefore, the acceptance probability can be computed as:

$$\begin{aligned} a(X'|X_{i-1}) &= \min\left\{1, \frac{I(X')T(X_{i-1}|X')}{I(X_{i-1})T(X'|X_{i-1})}\right\} \\ &= \min\left\{1, \frac{I(X')}{I(X_{i-1})}\right\}. \end{aligned} \quad (12)$$

**Recording the expected sample values:** We use the *expected values* method [VG97] to enhance the basic Metropolis algorithm. We record the flux of the photons along both paths  $X_{i-1}$  and  $X'$  to the corresponding regions using Equation (4) with different weights. The weights are  $(1 - a)$  for  $X_{i-1}$  and  $a$  for  $X'$ , where  $a$  is the acceptance probability. As a result, we can achieve a smoother rendered result.

### 5. Combination with Direct Lighting

When simulating direct illumination, the direct lighting method is usually more efficient than the SPPM approach (the first row of Figure 4). Our method can be combined with direct lighting for more efficiently rendering global illumination. When the scene with lighting is dominated by direct illumination, we would like to compute it using direct lighting in the distributed ray tracing pass, and use the stochastic radiance estimator to only approximate the indirect illumination. In this case, our Metropolis sampling method can adaptively trace more indirect photons to compute the indirect illumination, which is often highly inefficient to simulate using the unbiased methods, such as the caustic effects (the close-up in the second row of Figure 4). We can see that the percentage of the light paths that create visible indirect photons in the Torus scene (Figure 4) significantly increases from 4.6% to 72.9%. Therefore, the combined method is more efficient to simulate global illumination compared with the standard SPPM approach.



**Figure 4:** Torus scene rendered using our method combined with direct lighting. In the first row, direct lighting (the red close-up) is more efficient than the SPPM method (the green close-up) to simulate direct illumination using the same rendering time. In the second row, since Metropolis sampling traces significantly more indirect photons to simulate indirect illumination, the combined method produces higher quality results (the red close-up) compared with the standard SPPM method (the green close-up).

## 6. Results

We implemented our method in the PBRT system [PH10] by modifying the standard SPPM approach [HJ09] to use our Metropolis sampling technique. We focused on comparing our adaptive sampling method used in SPPM with the uniform random sampling in the same rendering time. Since our method holds the robustness of SPPM, readers can refer to [HOJ08, HJ09] for more comparisons with the unbiased methods and the normal photon mapping. All the test scenes were rendered on a 2.93 GHz Intel Core i7 X870 using one core. We used  $0.5M$  emitted photons per iteration in all the scenes except for the Killeroos and Villa scenes. In the Killeroos and Villa scenes, we instead use the same number of iterations but different numbers of emitted photons per iteration for comparisons. In this case, we still get more visible photons with our method in the same rendering time. Table 1 summarizes the statistics for the test scenes.

**Table 1:** Rendering statistics for our test scenes. Both uniform random sampling and our adaptive method used the same rendering time. In both columns of uniform sampling and our method, the left is the total number of iterations, and the right is the percentage of the visible light paths. For the first four scenes, we use the same number of emitted photons per iteration; and for the last two scenes, we use the same number of iterations but different numbers of emitted photons in both methods. Our method usually performs a smaller number of iterations or uses smaller numbers of emitted photons per iteration, since the average length of the visible paths is often greater than that of the whole paths.

Scene	Time/min	Uniform	Our Method
Room	120	960 / 5.0%	563 / 53.3%
Bulb	63	234 / 25.8%	185 / 73.5%
Torus	37	209 / 11.7%	104 / 72.9%
Cornell Box	26	142 / 82.1%	128 / 92.8%
Killeroos	35	128 / 22.4%	128 / 69.2%
Villa	653	625 / 0.014%	625 / 30.9%

Figure 1 shows a room with difficult visibility configurations. Only a small portion of the light paths can normally contribute to the rendered image. The uniform sampling method samples light paths without considering the viewpoint. As a result, it performs inefficiently since only 5.0% of the paths can trace photons to the visible regions. In contrast, the decisive advantage of our Metropolis sampling method is its ability to handle the difficult visibility settings. Due to the local exploration of the visible light paths, the percentage of the visible paths in our method increases to 53.3%. Therefore, our adaptive method used in SPPM generates visually smoother results compared with the uniform random sampling. Our method also has lower numerical error compared to uniform photon tracing in the same rendering time, as demonstrated in Figure 5. The graph in Figure 5

shows that our method converges to the ground truth faster than the uniform sampling using the same rendering time.

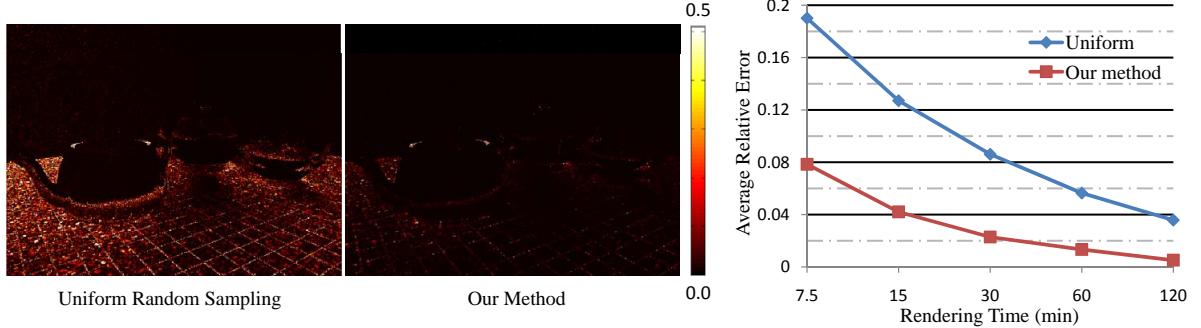
Figure 2 shows a bulb illuminating a substrate floor with bump mapping effects. The uniform sampling method traces photons without considering the scene features. Due to the complex occlusion, the photon density in some regions would be rather low (e.g., the photon density in the red highlighted areas is about 100 photons/pixel), causing the radiance estimator to become highly inefficient. Our Metropolis sampling method can also adapt to the local scene features. In our method, more paths are tailored to deposit photons in those problematic regions in the same rendering time (about 550 photons/pixel). In this way, our adaptive procedure can be done globally (i.e., the percentage of the visible paths increases from 25.8% to 73.5%), as well as locally, further improving the convergence speed of the SPPM approach.

In Figure 4 our method is combined with direct lighting to efficiently compute global illumination. Since direct lighting can efficiently simulate direct illumination, our method is merely used to compute indirect illumination. In this scene, the percentage of the light paths that create visible indirect photons dramatically increases from 4.6% to 72.9%. While the standard SPPM with uniform sampling uses 11.7% of the light paths to simulate global illumination. As a result, our method generates higher quality results compared with the standard SPPM method in the same rendering time.



**Figure 6:** Killeroos scene with motion blur and depth-of-field effects. Our method maintains the robustness of SPPM to simulate the distributed ray tracing effects, while significantly improving the rendering efficiency (the red close-up) where SPPM with uniform sampling tends to be inefficient (the green close-up) due to the insufficient photons.

Figure 6 shows that our method maintains the robustness of SPPM in simulation of the distributed ray tracing effects including motion blur and depth-of-field. We combine our method with direct lighting to render this scene. We use the same number of iterations in both methods. For comparisons in the same rendering time, we trace  $0.5M$  and  $1.1M$  emitted photons per iteration for our adaptive method and uniform sampling, respectively. The percentage of the paths that generate visible indirect photons increases from 4.7% to 69.2%



**Figure 5:** The numerical analysis of the rendering error in the Room scene. Left: color coded error images of Figure 1. Right: the average relative error of the progressive rendered results using uniform random sampling and our sampling method. The graph shows that our adaptive method converges to the ground truth faster than the uniform random sampling.

in our method. In contrast, only 22.4% of the total paths contribute to the rendered image using uniform sampling. Therefore, our improved method provides benefits to render complex global illumination as well as maintains the robustness of the standard SPPM approach.



**Figure 7:** Villa house illuminated indirectly by an infinite skylight and a discoid daylight outdoors. Using uniform sampling, only 0.014% of the light paths can arrive at the visible regions, causing SPPM to become extremely inefficient to render a smooth result (the green close-up). Thanks to the path reuse, the percentage of the visible light paths increases to 30.9% using our adaptive method, dramatically improving the rendering efficiency (the red close-up).

The Villa scene with complex geometry and illumination in Figure 7 is a particularly challenging scene for both biased and unbiased algorithms. There is no direct illumination inside the house, and all the light paths that arrive at the indoor regions must first follow specular bounces through the glass windows, making the unbiased methods extremely inefficient. Due to the difficult visibility settings, the SPPM method with uniform random sampling also becomes inefficient to render a high-quality smooth result, since only 0.014% of the light paths can create photons in the visible regions. In contrast, the percentage of the visible light paths is

dramatically improved to 30.9% using our Metropolis sampling strategy. For the equal time comparisons, we trace  $2M$  and  $28.6M$  emitted photons per iteration for our sampling method and uniform sampling, respectively. The result in Figure 7 demonstrates that our adaptive method used in SPPM significantly improves the rendering efficiency compared with the uniform sampling, especially for the complex scenes with difficult visibility settings.

## 7. Discussion

### 7.1. Validation of SPPM with adaptive sampling

One challenge for SPPM combined with Metropolis sampling is that the scalar contribution function changes as more samples are added. A visible light path would become invisible in the following iterations, because the shared search radii are gradually reduced. Therefore, it is difficult to ensure the unbiasedness since the precomputed integral  $b$  in Equation (6) may be inconsistent during the rendering process. To avoid this issue, we use a constant radius instead of the shared radius for each region to decide whether a light path is visible or not. In our implementation, the constant radius of each region is equal to the initial shared radius, which is approximate to a 3-pixel width using ray differentials.

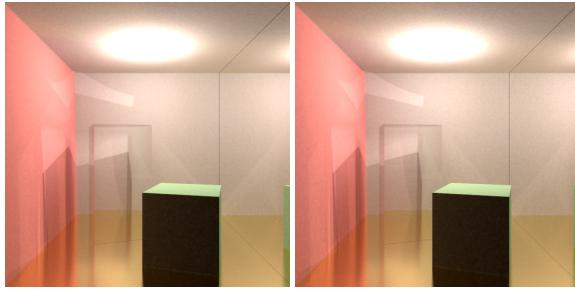
Another challenge is that the original SPPM method updates the statistics of photons based on the assumptions including using non-adaptive photon tracing. However, we show that the updating procedure of the statistics is still valid using our adaptive photon tracing algorithm in Appendix A. Therefore, we claim that the SPPM method combined with our Metropolis sampling is provably valid, which can also be verified in the experimental results.

### 7.2. Limitations

Compared with the recent adaptive photon tracing approach [HJ11], our sampling method can adapt to the local scene features, but at the cost of manual tweaking the sampling

parameters. A poor choice of the parameters could instead reduce the rendering efficiency. Nevertheless, the parameters are intuitive and simple to be tailored to the scene features. We can see that our sampling method is roughly equivalent to [HJ11] if we do not locally consider the photon distribution of the scenes (i.e., the scalar contribution function  $I(X) = 0$ , if  $X$  is invisible; and  $I(X) = 1$ , if  $X$  is visible).

Our adaptive method can improve the rendering efficiency when the photons are poorly distributed (especially for the case of difficult visibility settings). However, it does not provide benefits if most of the paths are visible and the photon distribution is roughly even, as shown in Figure 8. Our method produces similar quality results compared with the uniform random sampling, but at the cost of an additional small runtime overhead due to the precomputation.



**Figure 8:** Cornell box scene for image quality comparison between uniform random sampling (left) and our method (right). The results are practically identical as 82.1% of the paths are visible and the photon distribution is roughly even.

## 8. Conclusions and Future Work

We have presented an improvement to the SPPM approach [HJ09] by adopting a Metropolis sampling to optimize the photon distribution. In our method, we introduce a well-designed scalar contribution function as the Metropolis sampling strategy, effectively exploiting the local coherence among the light paths that trace photons to the visible regions. Using the same rendering time, the percentage of the visible light paths can be distinctly improved in our method. Furthermore, more photons can be locally traced to the problematic regions where the error tends to be large to improve the rendering efficiency. In addition, we would like to combine our method with direct lighting for further performance enhancements. The results show that our Metropolis sampling based method is more efficient than the standard SPPM using uniform random sampling for a wide range of scenes, while maintaining the robustness of the SPPM method in simulating complex global illumination.

Since the scalar contribution function in our Metropolis sampling is empirically based on the photon distribution, it may be unable to accurately estimate the rendering error.

For future work we would like to introduce a more accurate error estimation approach for SPPM, based on which we can design a more robust scalar contribution function as our Metropolis sampling strategy. Besides adaptively sampling the light paths, we would also like to adaptively sample the hit points in the distributed ray tracing pass for further performance enhancements. Finally, we expect that our adaptive sampling method can be used for other particle tracing algorithms, such as the normal photon mapping [Jen96], PPM [HOJ08] and the new formulation of PPM [KZ11], for improving the sampling efficiency.

## Acknowledgements

We would like to thank Yuxiang Wang and Peng Liu for discussions and the anonymous reviewers for their insightful comments and suggestions. The research was supported by Chinese 973 Program (2010CB328001), the National Science Foundation of China (61035002, 61003096) and Chinese 863 Program (2010AA186002). The third author was supported by Tsinghua University Initiative Scientific Research Program(2009THZ0) and the Fok Ying Tung Education Foundation (111070). The bulb scene is provided by Timothy James Armstrong. The torus scene is from Hachisuka *et al.* [HOJ08]. The room, the Killeroos, and the Villa scenes are from PBRT [PH10].

## References

- [CPC84] COOK R. L., PORTER T., CARPENTER L.: Distributed ray tracing. *SIGGRAPH Comput. Graph.* 18, 3 (1984), 137–145. [1](#)
- [CTE05] CLINE D., TALBOT J., EGBERT P.: Energy redistribution path tracing. *ACM Trans. Graph.* 24, 3 (2005), 1186–1195. [2](#)
- [CWW<sup>\*</sup>11] CHEN J., WANG B., WANG Y., OVERBECK R. S., YONG J.-H., WANG W.: Efficient depth-of-field rendering with adaptive sampling and multiscale reconstruction. *Comput. Graph. Forum* 30, 0 (2011), to appear. doi: 10.1111/j.1467-8659.2011.01854.x. [2](#)
- [DBBS06] DUTRE P., BALA K., BEKAERT P., SHIRLEY P.: *Advanced Global Illumination*. AK Peters Ltd, Natick, MA, USA, 2006. [1](#)
- [ETH<sup>\*</sup>09] EGAN K., TSENG Y.-T., HOLZSCHUCH N., DURAND F., RAMAMOORTHI R.: Frequency analysis and sheared reconstruction for rendering motion blur. *ACM Trans. Graph.* 28, 3 (2009), 93:1–93:13. [2](#)
- [FCL05] FAN S., CHENNEY S., LAI Y.-C.: Metropolis photon sampling with optional user guidance. In *Proceedings of Eurographics Symposium on Rendering* (2005), pp. 127–138. [2](#)
- [Has70] HASTINGS W. K.: Monte carlo sampling methods using markov chains and their applications. *Biometrika* 57, 1 (1970), 97–109. [2](#)
- [HH10] HOBEROCK J., HART J. C.: Arbitrary importance functions for metropolis light transport. *Comput. Graph. Forum* 29, 6 (2010), 503–513. [2](#)
- [HJ09] HACHISUKA T., JENSEN H. W.: Stochastic progressive photon mapping. *ACM Trans. Graph.* 28, 5 (2009), 141:1–141:8. [1, 2, 3, 6, 8, 9](#)

- [HJ11] HACHISUKA T., JENSEN H. W.: Robust adaptive photon tracing using photon path visibility. *ACM Trans. Graph.* 30, 0 (2011), in revision. <http://graphics.ucsd.edu/~toshiya/amcmppm.pdf>. 2, 7, 8
- [HJJ10] HACHISUKA T., JAROSZ W., JENSEN H. W.: A progressive error estimation framework for photon density estimation. *ACM Trans. Graph.* 29, 6 (2010), 144:1–144:12. 3
- [HJW\*08] HACHISUKA T., JAROSZ W., WEISTROFFER R. P., DALE K., HUMPHREYS G., ZWICKER M., JENSEN H. W.: Multidimensional adaptive sampling and reconstruction for ray tracing. *ACM Trans. Graph.* 27, 3 (2008), 33:1–33:10. 2
- [HOJ08] HACHISUKA T., OGAKI S., JENSEN H. W.: Progressive photon mapping. *ACM Trans. Graph.* 27, 5 (2008), 130:1–130:8. 1, 2, 6, 8
- [Jen96] JENSEN H. W.: Global illumination using photon maps. In *Proceedings of the eurographics workshop on Rendering techniques '96* (1996), pp. 21–30. 1, 2, 8
- [Kaj86] KAJIYA J. T.: The rendering equation. *SIGGRAPH Comput. Graph.* 20, 4 (1986), 143–150. 1, 2
- [KKK09] KITAOKA S., KITAMURA Y., KISHINO F.: Replica exchange light transport. *Comput. Graph. Forum* 28, 8 (2009), 2330–2342. 2
- [KSKAC02] KELEMEN C., SZIRMAY-KALOS L., ANTAL G., CSONKA F.: A simple and robust mutation strategy for the metropolis light transport algorithm. *Comput. Graph. Forum* 21, 3 (2002), 531–540. 2, 5
- [KZ11] KNAUS C., ZWICKER M.: Progressive photon mapping: A probabilistic approach. *ACM Trans. Graph.* 30, 3 (2011), 25:1–25:13. 8, 9
- [LFC07] LAI Y.-C., FAN S., CHENNEY S., CHARLES D.: Photorealistic image rendering with population monte carlo energy redistribution. In *Proceedings of Eurographics Symposium on Rendering* (2007), pp. 287–296. 2
- [LW93] LAFORTUNE E. P., WILLEMS Y. D.: Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Computographics' 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 machines. *The Journal of Chemical Physics* 21, 6 (1953), 1087–1092. 2
- [ODR09] OVERBECK R. S., DONNER C., RAMAMOORTHI R.: Adaptive wavelet rendering. *ACM Trans. Graph.* 28, 5 (2009), 140:1–140:12. 2
- [PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation (second edition)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. 5, 6, 8
- [SKDP99] SZIRMAY-KALOS L., DORNBACH P., PURGATHOFER W.: On the start-up bias problem of metropolis sampling. In *Proceedings of WSCG '99* (1999), pp. 273–280. 2
- [SSD\*09] SOLER C., SUBR K., DURAND F., HOLZSCHUCH N., SILLION F.: Fourier depth of field. *ACM Trans. Graph.* 28, 2 (2009), 18:1–18:12. 2
- [VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Computer Graphics (SIGGRAPH Proceedings)* (Los Angeles, USA, 1995), pp. 419–428. 2
- [VG97] VEACH E., GUIBAS L. J.: Metropolis light transport. In *Computer Graphics (SIGGRAPH Proceedings)* (Los Angeles, USA, 1997), pp. 65–76. 2, 3, 4, 5

**Appendix A:** Validity of updating photon statistics using our Metropolis sampling

According to the assumptions including using non-adaptive photon tracing, the following equation can be derived:

$$R_{i+1}(\vec{x}) = R_i(\vec{x}) \sqrt{\frac{N_i(\vec{x}) + \alpha M_i(\vec{x})}{N_i(\vec{x}) + M_i(\vec{x})}} = R_i(\vec{x}) C_P, \quad (13)$$

where  $C_P$  is independent of  $\vec{x}$ . Then the updating procedure of the photon statistics, i.e. Equations (3) and (5), can be derived based on Equation (13) (see [HJ09] for more details).

In our adaptive sampling method, we use a constant radius for each region to decide whether a light path is visible or not. That means the adaptive photon tracing is independent of the reduction of the shared search radius. Although using Metropolis sampling, we claim that the new created photon density for each iteration is approximately locally constant if a sufficient number of photons are traced in each iterations. According to the derivations in [KZ11], we can get

$$R_{i+1}(\vec{x}) = R_i(\vec{x}) \sqrt{\frac{i+\alpha}{i+1}} = R_i(\vec{x}) C'_P, \quad (14)$$

where  $C'_P = \sqrt{\frac{i+\alpha}{i+1}}$  is also independent of  $\vec{x}$ . This means that the updating procedure of photon statistics for SPPM using our adaptive sampling technique is still theoretically valid.