# Progressive Photon Mapping

Toshiya Hachisuka
UC San Diego

Shinji Ogaki
The University of Nottingham

Henrik Wann Jensen
UC San Diego

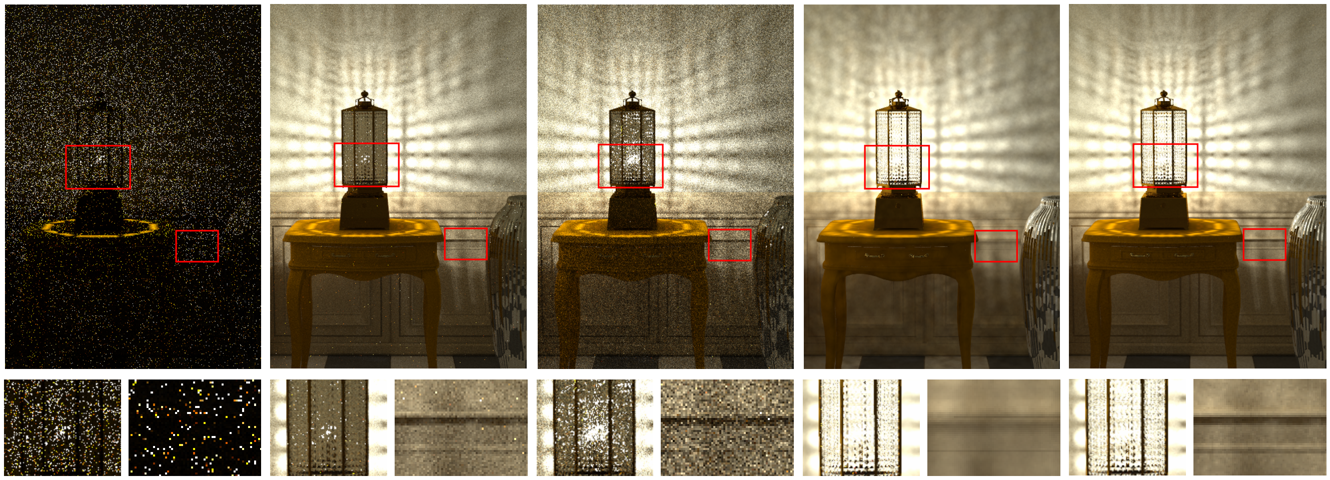



**Figure 1:** *A glass lamp illuminates a wall and generates a complex caustics lighting pattern on the wall. This type of illumination is difficult to simulate with Monte Carlo ray tracing methods such as path tracing, bidirectional path tracing, and Metropolis light transport. The lighting seen through the lamp is particularly difficult for these methods. Photon mapping is significantly better at capturing the caustics lighting seen through the lamp, but the final quality is limited by the memory available for the photon map and it lacks the fine detail in the illumination. Progressive photon mapping provides an image with substantially less noise in the same render time as the Monte Carlo ray tracing methods and the final quality is not limited by the available memory.*

## Abstract

This paper introduces a simple and robust progressive global illumination algorithm based on photon mapping. Progressive photon mapping is a multi-pass algorithm where the first pass is ray tracing followed by any number of photon tracing passes. Each photon tracing pass results in an increasingly accurate global illumination solution that can be visualized in order to provide progressive feedback. Progressive photon mapping uses a new radiance estimate that converges to the correct radiance value as more photons are used. It is not necessary to store the full photon map, and unlike standard photon mapping it possible to compute a global illumination solution with any desired accuracy using a limited amount of memory. Compared with existing Monte Carlo ray tracing methods progressive photon mapping provides an efficient and robust alternative in the presence of complex light transport such as caustics and in particular reflections of caustics.

**Keywords:** Global Illumination, Photon Mapping, Sampling and Reconstruction, Density Estimation

## 1 Introduction

Efficiently simulating global illumination is one of the classic problems in computer graphics. In the most general form it involves solving for all types of light transport within a scene in order to provide a full solution to the rendering equation [Kajiya 1986]. Several global illumination algorithms have been developed over the years and a number of algorithms based on Monte Carlo ray tracing are capable of solving the rendering equation without any approximations [Dutré et al. 2006].

Monte Carlo based methods can simulate both specular and diffuse materials, but there is one combination of these materials that is particularly problematic for most of the methods. This combination involves light being transported along a specular to diffuse to specular path (SDS path) before being seen by the eye. An example of an SDS path is the shimmering light seen on the bottom of a swimming pool, or it could be any type of specular surface seen in a room illuminated by a light source enclosed in glass. Most artificial lighting involves light sources embedded in glass (e.g. light bulbs, and headlights of car), and this type of illumination is very common. SDS paths are particularly challenging when the light source is small since the probability of sampling the light through the specular material is low in unbiased Monte Carlo ray tracing methods such as path tracing, bidirectional path tracing, and Metropolis light transport. For example, this can be seen in Figure 1 where a glass lamp illuminates a room. Notice how the Monte Carlo ray tracing methods fail to capture the transmission of caustics through the lamp.

Photon mapping is a consistent algorithm which is good at simulating caustics and SDS paths. However, photon mapping becomes very costly for scenes dominated by caustics illumination since the caustics are simulated by directly visualizing the photon map. To avoid noise it is necessary to use a large number of photons and the accuracy is limited by the memory available for the photon map. The fourth image in Figure 1 shows the lamp scene rendered using 20 million photons, but this is not enough to capture details in the illumination and the rendered image exhibits low-frequency noise.

In this paper we present a progressive photon mapping algorithm that makes it possible to robustly simulate global illumination including SDS paths with arbitrary accuracy without requiring infinite memory. Progressive photon mapping uses multiple photon tracing steps to compute an accurate solution without maintaining every photons from each iteration. We use a novel progressive radiance estimate that converges to the correct solution as more photons are added. The progressive radiance estimate uses the consistency of photon mapping to refine the estimate and ensure convergence. Our results show that progressive photon mapping is more efficient and robust than path tracing, bidirectional path tracing and Metropolis light transport in scenes with SDS paths. Furthermore, progressive photon mapping can compute a noise free solution in complex scenes where traditional photon mapping is limited by the memory available for the photons in the photon map.

## 2 Related Work

In his seminal paper on the rendering equation Kajiya [1986] introduced the path tracing algorithm. Path tracing is a Monte Carlo ray tracing algorithm, that computes global illumination in a given scene by evaluating a large number of random ray paths. This approach works well in scenes with smooth illumination, but it becomes costly in the presence of caustics due to small light sources. This is caused by the low probability of generating random paths that are reflected by the specular surface towards the light source.

To render scenes with caustics more efficiently Dutré et al. [1993] used Monte Carlo light tracing from the light sources to the image pixels. This approach renders caustics, but it cannot capture specular reflections seen by the observer. To address this issue Lafortune et al. [1993] and Veach and Guibas [1995] introduced bidirectional path tracing (BDPT). BDPT traces light paths both from the light source and the eye, which makes it significantly more efficient at rendering caustics. However, BDPT is very inefficient at rendering mirror reflections and transmissions of caustics, since it cannot connect the light path and the eye path in this case.

To address the shortcomings of BDPT Veach and Guibas [1997] proposed the Metropolis light transport algorithm (MLT). In MLT, each path is generated based on the mutation (perturbation) of a previous path. MLT can render complex illumination effects and it is particularly good at handling strongly localized illumination effects, such as illumination coming through a slight opening of door. However, in the case of mirror reflections of caustics, even MLT becomes inefficient because such paths are difficult to generate by mutating existing paths. Veach and Guibas introduced a special caustics mutation for this case, but it is only effective in scenes with large light sources.

Cline et al. proposed an improvement to MLT called Energy Redistribution Path Tracing (ERPT) [Cline et al. 2005]. ERPT is in essence a stratification of MLT over pixels, and it is simpler to implement than MLT. Since ERPT still depends on mutations similar to MLT, it shares the same weakness in the context of mirror reflections of caustics. To address this issue Cline et al. used image filtering to reduce noise. Unfortunately, the filter introduces an arbitrary amount of error in the resulting image. Recently, Lai et al. applied the Population Monte Carlo method to global illumination rendering (PMC-ER) [Lai et al. 2007]. They modified ERPT using Population Monte Carlo sampling. Their method is essentially an iterative importance sampling approach, and they demonstrated some improvement in efficiency over ERPT.

In his PhD thesis, Veach formulated unbiased Monte Carlo rendering methods as an integration over path space [Veach 1998]. Path space is the space of all possible light transport paths in a scene. He formulated unbiased Monte Carlo rendering methods as a Monte Carlo sampling from this path space. He also pointed out several limitations of unbiased path space based methods. One of the limitations is that any path space based method cannot render perfect specular reflections of caustics from a point light source viewed through a pinhole camera. The reason for this is that the probability of generating a path that connects the eye with the light is zero. Veach mentioned that changing the light source into a small area light addresses this problem, but as our results demonstrates such small area lights are still difficult to sample and contribute considerable noise to the final image. Note that path tracing, BDPT, MLT, ERPT and PMC-ER are all unbiased path-space based methods.

Photon mapping is a two-pass global illumination algorithm developed by Jensen [1996]. The first pass is building a photon map using photon tracing, and the second pass uses ray tracing to render the image. In the ray tracing pass the photon map is used to estimate the radiance at different locations within the scene. This is done by locating the nearest photons and performing a nearest neighbor density estimation. Since the density estimation process can be considered as a way of loosely connecting paths from the eye to the light, photon mapping is very effective at rendering SDS paths. The density estimation process effectively blurs the lighting in the scene, and to represent sharp illumination details it is necessary to use a large number of photons. The main challenge is scenes with lighting dominated by caustics. This type of lighting is rendered using a direct visualization of the photon map, which can require a large number of photons to resolve the details. Since the photon map is stored in memory, the final quality is often limited by the maximum number of photons that can be stored in the photon map. We overcome this problem in photon mapping by using a sequence of smaller photon maps without the need of storing all the photons.

To improve the performance of photon mapping and final gathering Havran et al. introduced the concept of reverse photon mapping [Havran et al. 2005]. Reverse photon mapping uses ray tracing in the first pass and photon tracing in the second pass. The ray tracing step builds a kd-tree over the hit points and in the photon tracing step this kd-tree is used to find the nearest hit points that a photon contributes to. The motivation for this approach is to reduce the complexity and improve the performance of photon mapping when a large number of rays are used in the ray tracing pass (e.g. as part of final gathering). We use a concept similar to reverse photon mapping, but our goal is to compute a highly accurate global illumination solution in scenes with complex lighting and to overcome the inherent limitation of the final image quality in standard photon mapping.

Suykens and Willems [2000] proposed an adaptive image filtering algorithm that asymptotically converges to the correct solutions. They described several heuristics that reduces the width of filtering kernel based on the number of samples. Our method also uses the number of photons to reduce the search radius of the radiance estimate, which asymptotically converges to the correct solutions similar to their method. The key difference is that our method utilizes the fact that photon mapping is a consistent method. Therefore we can keep the robustness of photon mapping, while obtaining asymptotically correct images.

A number of papers have addressed the issues in the standard photon mapping algorithm. Ray splatting [Herzog et al. 2007] removes boundary bias and topology bias [Schregle 2003] by using rays rather than photons to perform the radiance estimate. Proximity bias is not avoided as these methods still rely on nearest neighbor density estimation. Fradin et al. [2005] presented an out of core photon mapping approach optimized for large buildings where only a small part of the photon map is used during rendering. Christensen et al. [2004] introduced brick maps as a compact approximate representation of the illumination represented by the photon map. With progressive photon mapping we can use an unlimited number of photons, since we do not need to store all the photons and we retain all the advantages of the standard photon mapping method such as being able to handle non-Lambertian surfaces.

## 3 Overview

Photon Mapping [Jensen 2001] is a two-pass algorithm. The first pass is photon tracing, which traces photons from the light sources into the scene and stores them in a photon map as they interact with the surfaces. The second pass is rendering in which the photon map is used to estimate the illumination in the scene. Given a photon map, exitant radiance at any surface location $x$ can be estimated as:

$$L(x,\vec{\omega}) \approx \sum_{p=1}^{n} \frac{f_r(x,\vec{\omega},\vec{\omega}_p)\phi_p(x_p,\vec{\omega}_p)}{\pi r^2}\,, \qquad (1)$$

where $n$ is the number of nearest photons used to estimate the incoming radiance. $\phi_p$ is the flux of the $p$th photon, $f_r$ is the BRDF, $\vec{\omega}$ and $\vec{\omega}_p$ are the outgoing and incoming directions. $r$ is the radius of the sphere containing the $n$ nearest photons. This estimate assumes that the local set of photons represents incoming radiance at $x$, and that the surface is locally flat around $x$.

The radiance estimate in Equation 1 is the source of bias in photon mapping. The photon tracing step is unbiased, but the resulting photon distribution is blurred as part of the radiance estimate. As the photon density increases the radiance estimate will converge to the correct solution, and this makes photon mapping a *consistent* algorithm. To ensure convergence to the correct solution it is necessary to use an infinite number of photons in the photon map and in the radiance estimate. Furthermore, the radius should converge to zero. We can satisfy these requirements by using $N$ photons in the photon map, but only $N^\beta$ with $\beta \in ]0:1[$ photons in the radiance estimate. As $N$ becomes infinite both $N$ and $N^\beta$ will become infinite, but $N^\beta$ will be infinitely smaller than $N$, which ensures that $r$ will converge to zero. This can be written as [Jensen 2001]:

$$L(x,\vec{\omega}) = \lim_{N\to\infty} \sum_{p=1}^{\lfloor N^\beta \rfloor} \frac{f_r(x,\vec{\omega},\vec{\omega}_p)\phi_p(x_p,\vec{\omega}_p)}{\pi r^2}\,, \qquad (2)$$

In standard photon mapping this result is only of theoretical interest since all the photons are stored in memory. This makes it impossible to obtain a solution with arbitrary precision. In the following sections we describe a new radiance estimate that fulfills the requirements of Equation 2 without having to store all the photons in memory.

### 3.1 Progressive Photon Mapping

The main idea in progressive photon mapping is to reorganize the standard photon mapping algorithm based on the conditions of consistency, in order to compute a global illumination solution with arbitrary accuracy without storing the full photon map in memory. Progressive photon mapping is a multi-pass algorithm in which the first pass is ray tracing and all subsequent passes use photon tracing. Each photon tracing pass improves the accuracy of the global illumination solution and the algorithm is progressive in nature. Figure 2 summarizes our algorithm.

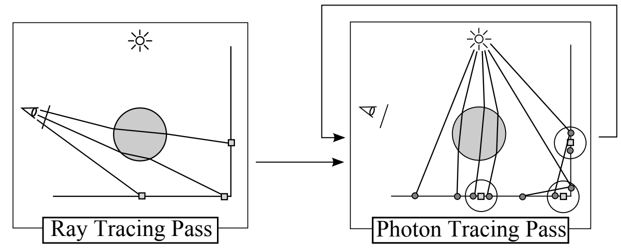



**Figure 2:** *Progressive photon mapping uses ray tracing in the first pass followed by one or more photon tracing passes.*

**Ray Tracing Pass:** The ray tracing pass is similar to reverse photon shooting [Havran et al. 2005] without the use of gathering. It uses standard ray tracing to find all the surfaces in the scene visible through each pixel in the image (or a set of pixels). Note, that each ray path includes all specular bounces until the first non-specular surface seen. In scenes with a large number of specular surfaces the length of the ray paths can be limited by using Russian Roulette. For each ray path we store all hit points along the path where the surface has a non-specular component in the BRDF. With each hit point we store the hit location $x$, the ray direction $\vec{\omega}$, scaling factors including BRDF and pixel filtering value, and the associated pixel location. In addition we store extra data necessary for the progressive radiance estimate including a radius, the intercepted flux, and the number of photons within the radius. We represent these values in the following structure:

```
struct hitpoint {
  position x       Hit location
  normal n         Normal at x
  vector ω         Ray direction
  integer BRDF     BRDF index
  float x,y        Pixel location
  color wgt        Pixel weight
  float R          Current photon radius
  integer N        Accumulated photon count
  color τ          Accumulated reflected flux
}
```

We will describe how to compute the last three values in more detail in the next section.

**Photon Tracing Passes:** The photon tracing step is used to accumulate photon power at the hit points found in the ray tracing pass. It can be divided into multiple passes where each pass consists of tracing a given number of photons into the scene in order to build a photon map. After each photon tracing pass we loop through all hit points (from the ray tracing pass) and find the photons within the radius of each hit point. We use the newly added photons to refine the estimate of the illumination within the hit point as described in the following section. Once the contribution of the photons have been recorded they are no longer needed, and we can discard all photons and proceed with a new photon tracing pass. This continues until enough photons have been accumulated and the final image quality is sufficient. Note that we can render an image after each photon tracing pass. As more photons are accumulated the quality of the image will progressively improve toward the final result.

## 4 Progressive Radiance Estimate

The traditional photon map radiance estimate as given in Equation 1 relies on an estimate of the local density of photons. The estimate of the local density $d(x)$ is:

$$d(x) = \frac{n}{\pi r^2} \,. \tag{3}$$

This estimate is based on locating the $n$ nearest photons within a sphere of radius $r$, assuming that the surface is locally flat such that the photons are located within a disc. If we generate another photon map and use it to compute the density at $x$ we might find $n'$ photons within the *same disc*, which may results in a different density estimate $d'(x)$:

$$d'(x) = \frac{n'}{\pi r^2} \,. \tag{4}$$

Note that we are using the same radius as Equation 3. By averaging $d(x)$ and $d'(x)$ we can obtain a more accurate estimate of the density within the disc of radius $r$. This approach was proposed by Christensen [Jensen et al. 2004], and it will lead to a smoother radiance estimate, but the final result does not have more detail than each individual photon map. Furthermore, the averaging procedure is not consistent and the method will not converge to the exact value at $x$. Instead it computes the average value within a constant radius $r$. As a result, it cannot resolve small details within the radius $r$, and the accuracy is effectively limited by the total number of photons in each individual photon map.

The progressive radiance estimate combines the result from several photon maps in such a way that the final estimate will converge to the correct solution. It is able to resolve details in the illumination that is not captured by the individual photon maps. The key insight that makes this possible is a new technique for reducing the radius in the radiance estimate at each hit point, while increasing the number of accumulated photons. This effectively ensures that the photon density becomes infinite in the limit in accordance with Equation 2. In the following sections we describe how the photon density is progressively increased. We perform the radiance estimate computation at each hit point generated in the ray tracing pass. Initially, the radius, $R(x)$, at $x$ is set to a non-zero value such as the footprint of the pixel. It is also possible to estimate the radius after the first photon tracing pass by using the photon map to estimate the radius around each hit point.

### 4.1 Radius Reduction

Each hit point has a radius, $R(x)$. Our goal is to reduce this radius while increasing the number of photons accumulated within this radius, $N(x)$. The density $d(x)$ at a hit point $x$ is computed using Equation 3. Assume that a number of photon tracing steps have been performed and that $N(x)$ photons have been accumulated at $x$. If we perform one additional photon tracing step and find $M(x)$ photons within the radius $R(x)$ then we can add these $M(x)$ photons to $x$, which results in a new photon density $\hat{d}(x)$:

$$\hat{d}(x) = \frac{N(x) + M(x)}{\pi R(x)^2} \,. \tag{5}$$

The next step of the algorithm is reducing the radius $R(x)$ by $dR(x)$. If we assume that the photon density is constant within $R(x)$, we can compute the new total number of photons $\hat{N}(x)$ within a disc of radius $\hat{R}(x) = R(x) - dR(x)$ as:

$$\hat{N}(x) = \pi \hat{R}(x)^2 \hat{d}(x) = \pi (R(x) - dR(x))^2 \hat{d}(x) \,. \tag{6}$$

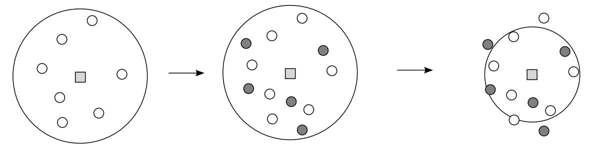



**Figure 3:** *Each hit point in the ray tracing pass is stored in a global data structure with an associated radius and accumulated photon power. After each photon tracing pass we find the new photons within the radius of each hit point, and we reduce the radius based on the newly added photons. The progressive radiance estimate ensures that the final value at each hit point will converge to the correct radiance value.*

To satisfy the consistency condition in Equation 2, there has to be a gain in the total number of photons at every iteration (i.e. $\hat{N}(x) > N(x)$). For simplicity, we use a parameter $\alpha = (0, 1)$ to control the fraction of photons to keep after every iteration. Therefore, $\hat{N}(x)$ can be computed as:

$$\hat{N}(x) = N(x) + \alpha M(x) \,, \tag{7}$$

which states that we would like to add $\alpha M(x)$ new photons at each iteration. We can compute the actual reduction of the radius $dR(x)$ by combining Equations 5, 6 and 7:

$$\begin{aligned} & \pi (R(x) - dR(x))^2 \hat{d}(x) = \hat{N}(x) \\ \Leftrightarrow \quad & \pi (R(x) - dR(x))^2 \frac{N(x) + M(x)}{\pi R(x)^2} = N(x) + \alpha M(x) \\ \Leftrightarrow \quad & dR(x) = R(x) - R(x) \sqrt{\frac{N(x) + \alpha M(x)}{N(x) + M(x)}} \,. \end{aligned} \tag{8}$$

Finally, the reduced radius $\hat{R}(x)$ is computed as:

$$\hat{R}(x) = R(x) - dR(x) = R(x) \sqrt{\frac{N(x) + \alpha M(x)}{N(x) + M(x)}} \,. \tag{9}$$

Note that this equation is solved independently for each hit point.

### 4.2 Flux Correction

When a hit point receives $M(x)$ photons we need to accumulate the flux carried by those photons. In addition we need to adjust this flux to take into account the radius reduction described in the previous section. Each hit point stores the unnormalized total flux received premultiplied by the BRDF. We call this quantity $\tau(x, \vec{\omega})$, and for the $N(x)$ photons it is computed as:

$$\tau_N(x, \vec{\omega}) = \sum_{p=1}^{N(x)} f_r(x, \vec{\omega}, \vec{\omega}_p) \phi'_p(x_p, \vec{\omega}_p) \,, \tag{10}$$

where $\vec{\omega}$ is the direction of the incident ray at the hit point, $\vec{\omega}_p$ is the direction of the incident photon, and $\phi'_p(x_p, \vec{\omega}_p)$ is unnormalized flux carried by the photon $p$. Note, that the flux at this stage is not divided by the number of emitted photons as in standard photon mapping. Similarly, the $M(x)$ new photons give:

$$\tau_M(x, \vec{\omega}) = \sum_{p=1}^{M(x)} f_r(x, \vec{\omega}, \vec{\omega}_p) \phi'_p(x_p, \vec{\omega}_p) \,. \tag{11}$$

If the radius was constant we could simply add $\tau_M(x, \vec{\omega})$ to $\tau_N(x, \vec{\omega})$, but since the radius is reduced we need to account for the photons that fall outside the reduced radius (see Figure 3). One method for finding those photons would be to keep a list of all photons within the disc and remove those that are not within the reduced radius disc. However, this method is not practical as it would require too much memory for the photon lists. Instead, we assume that the illumination and the photon density within the disc is constant, which results in the following adjustment:

$$
\begin{aligned}
\tau_{\hat{N}}(x, \vec{\omega}) &= (\tau_N(x, \vec{\omega}) + \tau_M(x, \vec{\omega})) \frac{\pi \hat{R}(x)^2}{\pi R(x)^2} \\
&= \tau_{N+M}(x, \vec{\omega}) \frac{\pi \left( R(x) \sqrt{\frac{N(x)+\alpha M(x)}{N(x)+M(x)}} \right)^2}{\pi R(x)^2} \\
&= \tau_{N+M}(x, \vec{\omega}) \frac{N(x) + \alpha M(x)}{N(x) + M(x)}, \qquad (12)
\end{aligned}
$$

where $\tau_{N+M}(x, \vec{\omega}) = \tau_N(x, \vec{\omega}) + \tau_M(x, \vec{\omega})$, and $\tau_{\hat{N}}(x, \vec{\omega})$ is the reduced value for the reduced radius disc corresponding to $\hat{N}(x)$ photons. The assumption that the photon density and thereby the illumination is constant within the disc may not be correct initially, but it becomes increasingly true as the radius becomes smaller except for points exactly at illumination discontinuities. This is not an issue, since the illumination at discontinuities is undefined and the probability of having a hit point exactly at a discontinuity is zero.

### 4.3 Radiance Evaluation

After each photon tracing pass we can evaluate the radiance at the hit points. Recall that the quantities stored include the current radius and the current intercepted flux multiplied by the BRDF. The evaluated radiance is multiplied by the pixel weight and added to the pixel associated with the hit point. To evaluate the radiance we further need to know the total number of emitted photons $N_{emitted}$ in order to normalize $\tau(x, \vec{\omega})$. The radiance is evaluated as follows:

$$
\begin{aligned}
L(x, \vec{\omega}) &= \int_{2\pi} f_r(x, \vec{\omega}, \vec{\omega}') L(x, \vec{\omega}') (\vec{n} \cdot \vec{\omega}')\, d\omega' \\
&\approx \frac{1}{\Delta A} \sum_{p=1}^{n} f_r(x, \vec{\omega}, \vec{\omega}_p) \Delta\phi_p(x_p, \vec{\omega}_p) \\
&= \frac{1}{\pi R(x)^2} \frac{\tau(x, \vec{\omega})}{N_{emitted}}. \qquad (13)
\end{aligned}
$$

Similar to the normal photon mapping, this formulation is not restricted to Lambertian materials as we premultiply the flux with the BRDF and store it as $\tau(x, \vec{\omega})$. Note that the radius $R(x)$ will not be reduced if the disc defined by $R(x)$ is within an unlit region (i.e., $M(x) = 0$). Although this situation seemingly breaks the conditions of consistency, it still converges to the correct solution $L(x, \vec{\omega}) = 0$, since $\tau(x, \vec{\omega})$ will not increase and $L(x, \vec{\omega}) \rightarrow 0$ as $N_{emitted} \rightarrow \infty$. Although the formal analysis of convergence properties of $R(x)$ and $N(x)$ has not been done yet, the graphs in Figure 4 indicate that the progressive radiance estimate converges to the correct radiance value $L(x)$, while the radius $R(x)$ is reduced to zero, and the number of photons $N(x)$ grows to infinity. The progressive radiance estimate ensures that the photon density at each hit point increases at each iteration and it is therefore consistent in accordance with Equation 2.

## 5 Results and Discussion

In this section we present results based on our implementation of the progressive photon mapping algorithm (PPM). For comparisons we have implemented path tracing (PT) with explicit direct light source sampling, bidirectional path tracing (BDPT) with multiple importance sampling [Veach and Guibas 1995], Metropolis light transport [Veach and Guibas 1997] based on the primary sample space [Kelemen et al. 2002] (MLT), and photon mapping (PM) [Jensen 2001]. The algorithms have been implemented in Pascal (Delphi), and all our examples have been rendered on a PC with 1GB of memory and a 2.4GHz Intel Core 2 Q6600 using one core. The rendered images have a width of 640 pixels. We used 100000 photons per photon shooting pass and $\alpha = 0.7$. We limited the standard photon mapping results to 20 million photons due to memory constraints on the machine we used for rendering. We used 500-1500 photons in the radiance estimate for our results. The exact number was picked to reduce low-frequency noise in the final image. All results are equal time comparisons except for photon mapping, where the rendering time is shorter since the total number of photons is limited by the available memory. Table 1 summarizes the computational effort for the different scenes.

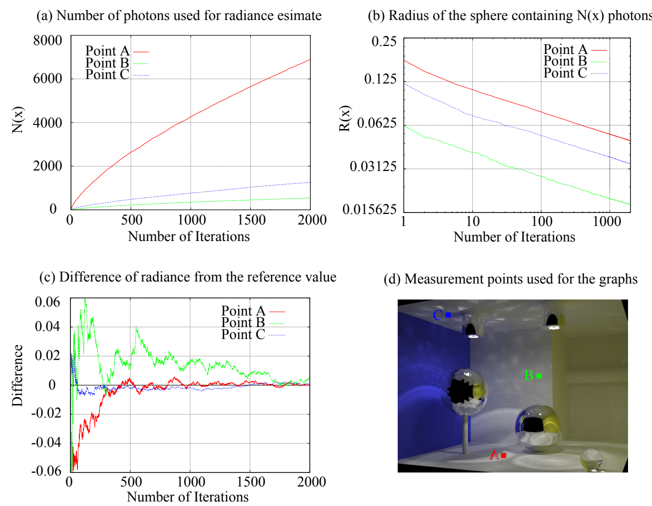
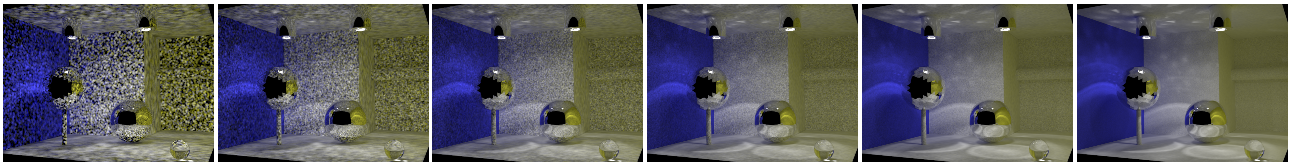



**Figure 4:** *Statistics at the hit points as a function of the number of iterations. The three hit points A, B, and C are indicated in (d). Each iteration is using 100000 photons. Note that the graph of $R(x)$ uses log scale for both axes.*

Figure 1 shows a glass lamp illuminating a wall. The images have been rendered using PT, BDPT, MLT, PM, and PPM. The image was rendered in 22 hours. Note, how the illumination seen in the glass lamp is very noisy in all the Monte Carlo ray tracing methods, while the photon mapping result exhibit low frequency noise even with 20 million photons. With progressive photon mapping we used 165 million photons, which captures the sharp features in the illumination without the high frequency noise seen in the Monte Carlo ray tracing images.

The test scene shown in Figure 6 is a box with two mirror balls and a smaller glass ball. One of the mirror balls is faceted to cause interesting caustics pattern. The scene is illuminated by two lighting fixtures, where each lighting fixture is a spherical light source behind a spherical lens inside the metal cylinders at the ceiling. This type of caustics illumination results in numerous SDS paths within the model. Note, how the reflections in the mirror ball are very noisy in both BDPT and MLT. Normal photon mapping results in a slightly blurry result. With progressive photon mapping we can increase the number of photons to 213 million, which makes it possible to capture the detailed illumination within the scene with considerable less noise than the Monte Carlo ray tracing methods. The Monte Carlo ray tracing images and the progressive photon



0.1M photons | 0.4M photons | 1.6M photons | 6.4M photons | 25.6M photons | 102.4M photons

**Figure 5:** *Sequence of images with increasing number of photons. The images show a direct visualization the progressive radiance estimate after 1, 16, 64, 256, and 1024 photon tracing passes. Note, how the illumination is quite good already after 0.1 million photons. Adding more photons makes the image sharper and reduces low frequency noise.*

| | PT Samples | BDPT Samples | MLT Mutations | PPM Iterations | Time Hours |
|---|---|---|---|---|---|
| Lamp | 840 | 80 | 82 | 1651 | 22 |
| Box | 1428 | 155 | 132 | 2134 | 4 |
| Torus | 1050 | 550 | 359 | 520 | 2 |
| Bathroom | 675 | 66 | 66 | 6126 | 16 |

**Table 1:** *Rendering statistics for the different scenes. The samples and mutations numbers are the average number per pixel. PPM uses 100000 photons per iteration.*

mapping image were rendered in 4 hours, while the normal photon mapping image was rendered in 1 hour. The progressive nature of our method is demonstrated on the box scene in Figure 5 where the intermediate results have been visualized.

Figure 7 demonstrates the illumination on a torus embedded in a glass cube (inspired by a similar scene by Cline et al. [2005]). All the Monte Carlo ray tracing methods have trouble rendering the lighting on the torus, while progressive photon mapping renders a smooth noise free result in the same time (2 hours). The reference image rendered using path tracing still exhibit noise even after using 51500 samples per pixel and 91 hours of rendering time.

The bathroom scene in Figure 8 is an example of complex geometry and illumination. The illumination in this scene is due to two small spherical area light sources enclosed within bumpy glass tubes. This type of illumination is common in real world lighting. Both BDPT and MLT fail to properly sample the reflection in the mirror and on the chrome pipes as well as the lighting behind the glass door. Our method robustly handles all of these illumination paths robustly. Normal photon mapping is also fairly robust in this scene, but it is not possible to render a noise free image using 20 million photons.

## 6 Conclusion and Future Work

We have presented a progressive refinement extension to photon mapping that makes it possible to compute solutions with arbitrary accuracy in scenes with complex illumination. Our results show that progressive photon mapping is particularly robust in scenes with complex caustics illumination, and it is more efficient than methods based on Monte Carlo ray tracing such as bidirectional path tracing and Metropolis light transport. The primary contribution is a new progressive radiance estimate that converges to the correct solution as more photons are added.

We believe progressive photon mapping opens several opportunities for future research. First, we would like to develop a stopping criteria and an error estimate based on the local statistics of the photon map around each hit point. This would address one of the outstanding problems in photon mapping, which is how to determine the number of photons required in a given scene. We would also like to explore adaptive photon tracing techniques, which could be effective in the progressive photon mapping framework since we know precisely how much each photon path contributes to the final image. It could also interesting to utilize statistics from the accumulated photons to determine optimal values for the parameters, such as $\alpha$ and the initial radius.

### Acknowledgments

This work was supported in part by NSF grant CPA 0701992. The glass lamp and bathroom models are courtesy of Youichi Kimura (Studio Azurite). The box scene is inspired by a similar scene used by WinOSi (http://www.winosi.onlinehome.de/).

## References

CHRISTENSEN, P. H., AND BATALI, D. 2004. An irradiance atlas for global illumination in complex production scenes. In *Proceedings of Eurographics Symposium on Rendering 2004*, 133–141.

CLINE, D., TALBOT, J., AND EGBERT, P. 2005. Energy redistribution path tracing. *ACM Trans. Graph. (SIGGRAPH Proceedings) 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 '93*, 128–137.

DUTRÉ, P., BEKAERT, P., AND BALA, K. 2006. *Advanced Global Illumination (2nd edition)*. A K Peters.

FRADIN, D., MENEVEAUX, D., AND HORNA, S. 2005. Out of core photon-mapping for large buildings. In *Proceedings of Eurographics Symposium on Rendering 2005*, Eurographics.

HAVRAN, V., HERZOG, R., AND SEIDEL, H.-P. 2005. Fast final gathering via reverse photon mapping. *Computer Graphics Forum (Proceedings of Eurographics 2005) 24*, 3 (September).

HERZOG, R., HAVRAN, V., KINUWAKI, S., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2007. Global illumination using photon ray splatting. In *Eurographics 2007*, Blackwell, vol. 26 of *Computer Graphics Forum*, 503–513.

JENSEN, H. W., SUYKENS, F., CHRISTENSEN, P., AND KATO, T. 2004. A practical guide to global illumination using ray tracing and photon mapping. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, ACM, New York, NY, USA, 20.

JENSEN, H. W. 1996. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques '96*, Springer-Verlag, London, UK, 21–30.

JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Ltd., Natick, MA.

KAJIYA, J. T. 1986. The rendering equation. *Computer Graphics (SIGGRAPH Proceedings) 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. *Computer Graphics Forum (Eurographics) 21*, 3, 531–540.

LAFORTUNE, E. P., AND WILLEMS, Y. D. 1993. Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics '93)*, H. P. Santo, Ed., 145–153.

LAI, Y.-C., FAN, S. H., CHENNEY, S., AND DYER, C. 2007. Photorealistic image rendering with population monte carlo energy redistribution. In *Proceedings of the Rendering Techniques (EGSR)*, Eurographics Association, Grenoble, France, 287–295.

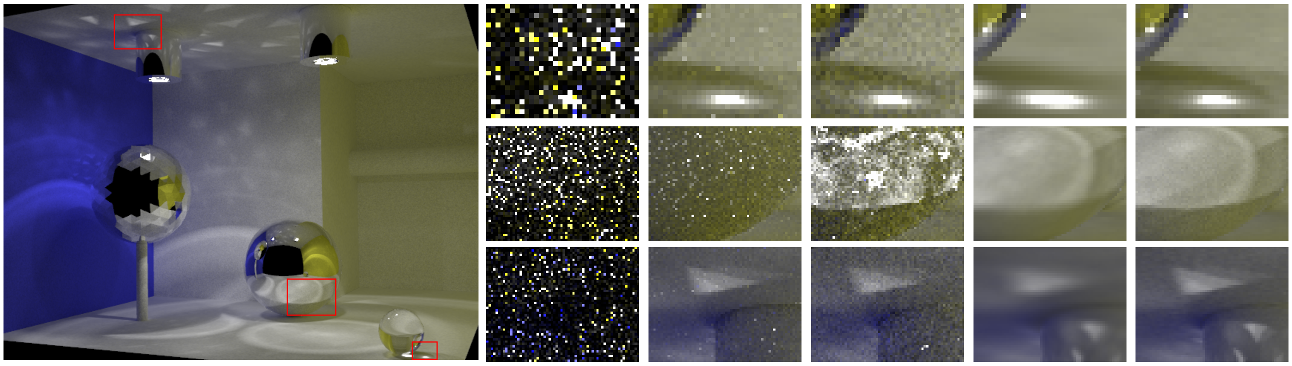
SCHREGLE, R. 2003. Bias compensation for photon maps. In *Computer Graphics Forum 22, 4 (2003), C792-C742*.

SUYKENS, F., AND WILLEMS, Y. D. 2000. Adaptive filtering for progressive monte carlo image rendering. In *Eighth International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG 2000)*.

VEACH, E., AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for monte carlo rendering. In *Computer Graphics (SIGGRAPH Proceedings)*, 419–428.

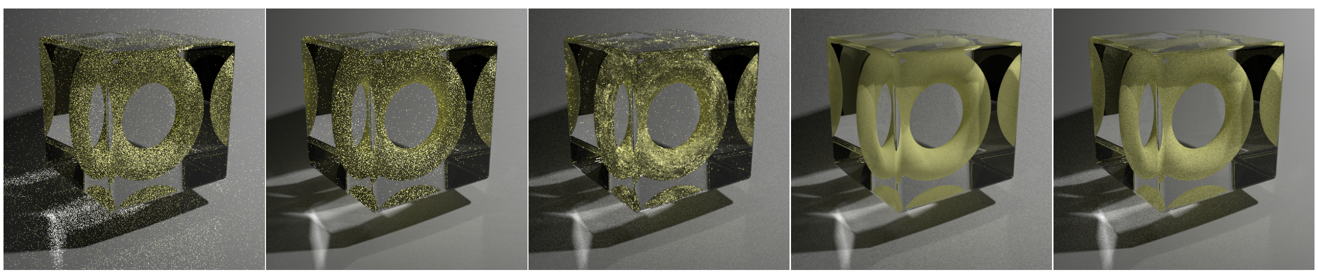
VEACH, E., AND GUIBAS, L. J. 1997. Metropolis light transport. In *Computer Graphics (SIGGRAPH Proceedings)*, 65–76.

VEACH, E. 1998. *Robust monte carlo methods for light transport simulation*. PhD thesis, Stanford, CA, USA. Adviser-Leonidas J. Guibas.



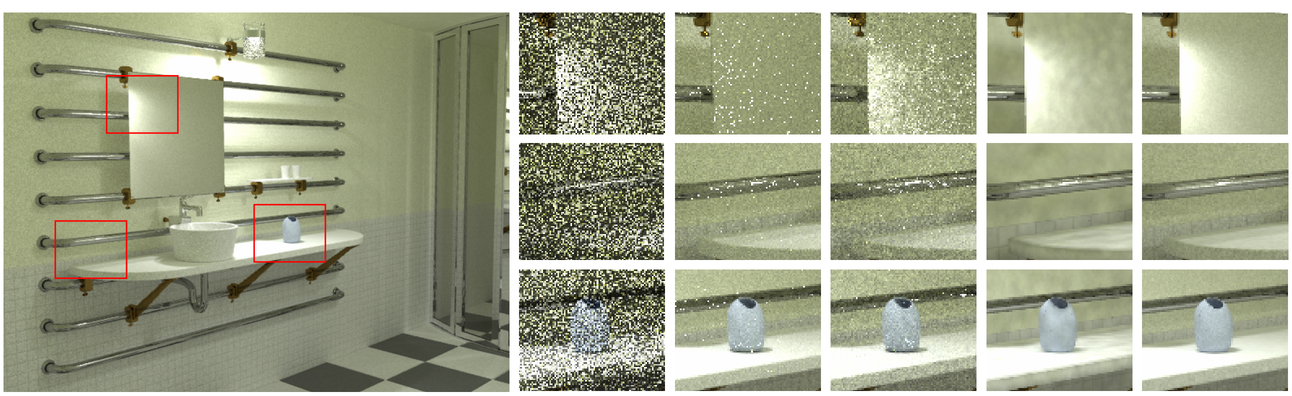
Progressive photon mapping | PT | BDPT | MLT | PM | PPM

**Figure 6:** *A box scene illuminated by a lighting fixture. The lighting fixture is behind glass and the illumination in the scene is dominated by caustics. The specular reflections and refractions have significant noise even with Metropolis light transport. Standard photon mapping cannot resolve the sharp illumination details in the scene with the maximum 20 million photons in the photon map. With progressive photon mapping we could use 213 million photons, which resolves all the details in the scene and provides a noise free image in the same rendering time as the Monte Carlo ray tracing methods.*



PT | BDPT | MLT | PPM | Reference

**Figure 7:** *Torus embedded in a glass cube. The reference image on the far right have been rendered using path tracing with 51500 samples per pixel. The Monte Carlo ray tracing methods fail to capture the lighting within the glass cube, while progressive photon mapping provides a smooth result using the same rendering time.*



Progressive photon mapping | PT | BDPT | MLT | PM | PPM

**Figure 8:** *Lighting simulation in a bathroom. The scene is illuminated by a small lighting fixture consisting of a light source embedded in glass. The illumination in the mirror cannot be resolved using Monte Carlo ray tracing. Photon mapping with 20 million photons results in a noisy and blurry image, while progressive photon mapping is able to resolve the details in the mirror and in the illumination without noise.*