

Progressive Photon Mapping: A Probabilistic Approach

Claude Knaus and Matthias Zwicker

University of Bern, Switzerland

In this paper we present a novel formulation of progressive photon mapping. Similar to the original progressive photon mapping algorithm, our approach is capable of computing global illumination solutions without bias in the limit, and it uses only a constant amount of memory. It produces high quality results in situations that are difficult for most other algorithms, such as scenes with realistic light fixtures where the light sources are completely enclosed by refractive material. Our new formulation is based on a probabilistic derivation. The key property of our approach is that it does not require the maintenance of local photon statistics. In addition, our derivation allows for arbitrary kernels in the radiance estimate and includes stochastic ray tracing algorithms. Finally, our approach is readily applicable to volumetric photon mapping. We compare our algorithm to previous progressive photon mapping approaches and show that we achieve the same convergence to unbiased results, even without local photon statistics.

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

Additional Key Words and Phrases: Global illumination, photon mapping

1. INTRODUCTION

Photon mapping [Jensen 2001] is one of the most popular algorithms to numerically approximate solutions of the rendering equation [Kajiya 1986]. It is based on Monte Carlo integration, similar to related algorithms such as path tracing [Kajiya 1986] and its variants [Lafortune and Willem 1993] or Metropolis light transport [Veach and Guibas 1997]. One of the main advantages of photon mapping is that, at equal computational cost, it can often produce images with less noise than other Monte Carlo algorithms. Photon mapping is consistent, in the sense that the numerical approximation converges to an exact solution as the number of Monte Carlo samples goes to infinity. In contrast to other Monte Carlo techniques, however, it is biased, which means that the expected error of any approximation with a limited number of samples is non-zero.

Authors' addresses: {knaus, zwicker}@iam.unibe.ch.

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.

© YYYY ACM 0730-0301/YYYY/11-ARTXXX \$10.00

DOI 10.1145/XXXXXX.YYYYYYYY

<http://doi.acm.org/10.1145/XXXXXX.YYYYYYYY>

The reason for the computational efficiency of photon mapping is that it caches and reuses Monte Carlo samples. In a first stage of the algorithm it caches the samples, or photons, in a spatial datastructure, the photon map. In a second stage these samples are reused in an approximation procedure, called radiance estimation, which basically counts the number of photons per circular area with a certain radius. This approximation, however, acts similarly to a low-pass filter on the cached samples. It always returns an overly smooth approximation of the true radiance, and hence causes the non-zero expected error, or bias, of the solution. This bias only vanishes in theory, if it were possible to cache an infinite number of photons.

Hachisuka et al. [2008; 2009] recently presented progressive photon mapping (PPM), a simple strategy that breaks this memory bottleneck. They incrementally update a sequence of photon mapping results, where each step in the sequence uses a limited number of photons. Over this sequence, the radiance estimation radius is reduced in each step. The key is to reduce the radius such that, in the limit, the incremental updates converge to an exact, unbiased solution of the rendering equation. Hachisuka et al. achieve this by maintaining local statistics for each region where a radiance estimate needs to be evaluated. The statistics include for example the number of photons collected in the region. In the simplest case, the regions are the points seen through each pixel. In stochastic PPM [Hachisuka and Jensen 2009], the regions are generalized to render effects such as glossy reflections or depth of field.

In this paper, we introduce a probabilistic derivation of progressive photon mapping. The key property of our approach is that it does not require the maintenance of local statistics. Therefore, we could call our approach memoryless progressive photon mapping. We show that each step in the sequence of photon mapping results can be performed completely independently. As a benefit, we can compute each step in parallel or with a standard photon mapper used as a black box. In addition, our derivation allows for arbitrary kernels in the radiance estimate. We also present an asymptotic convergence analysis that reveals the trade-off between vanishing variance and expected error, which is controlled using a single parameter. Our approach includes the scenario of stochastic progressive photon mapping in a simple and straightforward manner. Finally, we demonstrate that it is readily applicable to volumetric photon mapping, which has not been shown before. We compare our algorithm to previous progressive photon mapping approaches and show that we achieve the same convergence to unbiased results, even without local statistics.

In summary, we make the following contributions:

- a novel derivation of progressive photon mapping that is based on a probabilistic framework and includes stochastic PPM,
- an asymptotic convergence analysis yielding convergence rates for variance and expected error,
- a memoryless algorithm that does not require the maintenance of statistics, computes each step independently and if desired in parallel, works with arbitrary kernels, and is trivial to extend to volumetric photon mapping.

2. PREVIOUS WORK

Realistic rendering in computer graphics is governed by the rendering equation [Kajiya 1986]. Most standard algorithms to approximate solutions of the rendering equation, such as path tracing [Kajiya 1986], bi-directional path tracing [Lafortune and Willem 1993; Veach and Guibas 1995], photon mapping [Jensen 2001; Jensen and Christensen 1998], or Metropolis light transport [Veach and Guibas 1997; Pauly et al. 2000; Cline et al. 2005] are based on Monte Carlo integration. These methods are popular because of their conceptual simplicity, their generality, and because they lead to high quality results if enough Monte Carlo samples are evaluated.

Monte Carlo methods randomly sample light paths that connect a point on a light source to a point on the image sensor. The various algorithms differ mainly in the way these random paths are generated. Unbiased methods such as path tracing and its variations and Metropolis light transport always sample complete paths connecting a light source and the sensor. Biased methods such as photon mapping include an interpolation step, known as radiance estimation, to transport light between incomplete sub-paths. The disadvantage of biased methods is that they lead to non-zero expected error, which manifests itself as blurriness in the results. On the other hand, there are certain types of light paths that are very challenging to sample using unbiased methods. These are paths where emitted light is transported via specular surfaces to a diffuse surface, and from there again via specular surfaces to the sensor. In fact, if the aperture of the sensor is a pinhole and the source is a point light, then the probability to sample such a path is zero in any unbiased method. Although this scenario is not physically plausible, if the light and sensor aperture are small, it may still require an impractical number of samples to obtain an acceptable solution [Veach 1998]. Hachisuka et al. [2008] point out common examples for such situations, such as sunlight seen on the bottom of a swimming pool, or specular surfaces in a room illuminated by a light source enclosed in glass, which is how many types of light fixtures are constructed.

Photon mapping [Jensen 2001] is a biased method that often leads to results with less obvious artifacts in such situations, at the cost of non-zero expected error. The key idea in photon mapping is to sample sub-paths starting from light sources and cache these samples, or photons, in a spatial data structure. In a second step, it samples sub-paths from the sensor and connects them to the light sub-paths on non-specular surfaces. This is achieved using radiance estimation, which acts on the cached photons similarly as a smooth reconstruction filter. To avoid this smoothing effect and obtain unbiased results, however, it would be necessary to sample and store infinitely many light sub-paths. There are a number of improvements of the standard radiance estimation approach that attempt to reduce its bias [Schregle 2003; Herzog and Seidel 2007; Herzog et al. 2007; Spencer and Jones 2009]. Radiance estimation has also been extended to volumetric media [Jensen and Christensen 1998; Jarosz et al. 2008], and to time dependent photon mapping [Camarano and Jensen 2002]. But all these techniques still suffer from the same fundamental limitation that infinitely many photons need to be stored to obtain unbiased solutions.

Our work is inspired by progressive photon mapping (PPM) [Hachisuka et al. 2008] and the more recent stochastic PPM approach [Hachisuka and Jensen 2009]. These techniques have shown how to break the memory bottleneck of photon mapping. In theory, they can achieve unbiased results under

limited memory resources. Concurrent to our work, this approach has also been generalized to include arbitrary smooth kernels for radiance estimates [Hachisuka et al. 2010]. Here we derive similar algorithms but based on a more general, probabilistic framework. Our approach has the advantages that it does not require the maintenance of local statistics, it works with arbitrary kernels, and it is readily applicable to other radiance estimates such as volumetric radiance estimation. Although this is not impossible with the original PPM approach, it has not been shown yet. We also believe our approach leads to simpler algorithms because it does not require the storage of local statistics.

3. PRELIMINARIES: VARIANCE AND EXPECTED ERROR OF RADIANCE ESTIMATION

In this section we analyze the variance and expected error of radiance estimation in photon mapping from a probabilistic perspective. You find an overview of our notation in Table I. These results will form the basis for our derivation in the next section. Radiance estimation approximates the reflected radiance $L(x, \omega)$, where x represents a surface location and ω an outgoing direction, by computing a local weighted average of incident radiance multiplied with the BRDF [Veach 1998; Pharr and Humphreys 2004]. Photon mapping computes a Monte Carlo estimate of this measurement,

$$L(x, \omega) \approx \frac{1}{M} \sum_{j=1}^M k_r(x_j - x) \gamma_j. \quad (1)$$

The sum is over all M photons that were emitted in the scene, and x_j denote the photon positions. The kernel k_r is a function that determines the weight of each photon in the local average. It is related to a canonic kernel k via a linear scaling factor r that specifies the size of the local averaging window. For any r , the kernel has unit integral and depends on the difference $x_j - x$ between photons positions x_j and the location of the estimate x . The photon positions are generated using a random process, i.e., Monte Carlo particle tracing. They are distributed over the scene surfaces according to a certain probability density, which we will denote $p_l(x)$. Finally, γ_j is the contribution of each photon. This includes the product of the BRDF and the photon value. The photon value represents the contribution of the path traced out by the photon, divided by the probability with which the path was sampled.

We now analyze the variance and expected value of the error $\epsilon(x, r)$, which we define as the difference between the radiance estimate at position x and using a kernel k_r of scale r , and the true radiance $L(x, \omega)$,

$$\epsilon(x, r) = \frac{1}{M} \sum_{j=1}^M k_r(x_j - x) \gamma_j - L(x, \omega). \quad (2)$$

Variance. We denote the variance of $\epsilon(x, r)$ by $\text{Var}[\epsilon(x, r)]$. To obtain an estimate for this variance, we make the assumption that the probability density of the photons is constant within the support of the kernel k_r . Let this density be $p_l(x)$. We also interpret the photon values γ_j as samples of a random variable γ . This allows us to include the variance of photon contributions due to different photon paths and BRDF values on non-diffuse surfaces in our analysis. We show in Appendix A that, under our assumptions, the variance

x, ω	Surface point and outgoing direction
L	Reflected radiance
M	Number of emitted photons
x_j, γ_j	Position and contribution of photon j
k	Canonic kernel for radiance estimation
k_r	Kernel for radiance estimation with radius r
ϵ	Error of radiance estimation
$\text{Var} [\epsilon]$	Variance of error of radiance estimation
$E [\epsilon]$	Expected error of radiance estimation
p_l	Probability distribution of photons
N	Number of progressive photon mapping steps
c, \bar{c}_N	Exact pixel value, estimate after N steps
x_i, ω_i	Pos. and dir. of eye path hit point in step i
p_e	Probability distribution of eye path hit points
W	Contribution of eye paths
$\bar{\epsilon}_N$	Avg. error of radiance estimates after N steps
$\text{Var} [\bar{\epsilon}_N]$	Variance of average error of radiance estimates
$E [\bar{\epsilon}_N]$	Expected error of radiance estimates

Table I. : Overview of symbols.

can be approximated as

$$\text{Var} [\epsilon(x, r)] \approx \frac{(\text{Var} [\gamma] + E [\gamma]^2)p_l(x)}{M r^2} \int_{\mathbb{R}^2} k(\psi)^2 d\psi. \quad (3)$$

This confirms the intuition that variance decreases linearly with the number of emitted photons M and the square of the kernel scale. In addition, the above equation reveals the influence of the shape of the canonic kernel k on the variance. We can also specify a desired variance $\text{Var} [\epsilon]$, and obtain the corresponding kernel radius

$$r(x, \text{Var} [\epsilon]) \approx \sqrt{\frac{(\text{Var} [\gamma] + E [\gamma]^2)p_l(x)}{M \text{Var} [\epsilon]} \int_{\mathbb{R}^2} k(\psi)^2 d\psi}. \quad (4)$$

We will use this relationship later to derive the radii for our progressive photon mapping approach.

Expected Error. We show in Appendix B that the expected error of the radiance estimate $E [\epsilon(x, r)]$ is proportional to the squared radius, i.e.,

$$E [\epsilon(x, r)] = r^2 E [\gamma] \tau \quad (5)$$

for some constant τ . This confirms the intuition that the expected error, or bias, decreases proportionally to the squared scale of the kernel. Unfortunately, it is not easy to estimate the factor τ numerically, because it depends on higher order derivatives of the photon density distribution $p_l(x)$. However, this result will still be useful in our analysis in the next section.

Volumetric Radiance Estimate. Our analysis of variance and expected error can also be applied to the volumetric radiance estimate. This will allow us to easily extend progressive photon mapping to participating media, as shown in the next section. The volumetric radiance estimate has the same form as Equation 1, except that the kernel is three- instead of two-dimensional. Therefore, under the same assumptions as above, it is straightforward to perform an analogous analysis for the volumetric case. The result of the analysis is identical to Equations 3 to 5, up to replacing factors r^2 by r^3 and integration over \mathbb{R}^2 by integration over \mathbb{R}^3 .

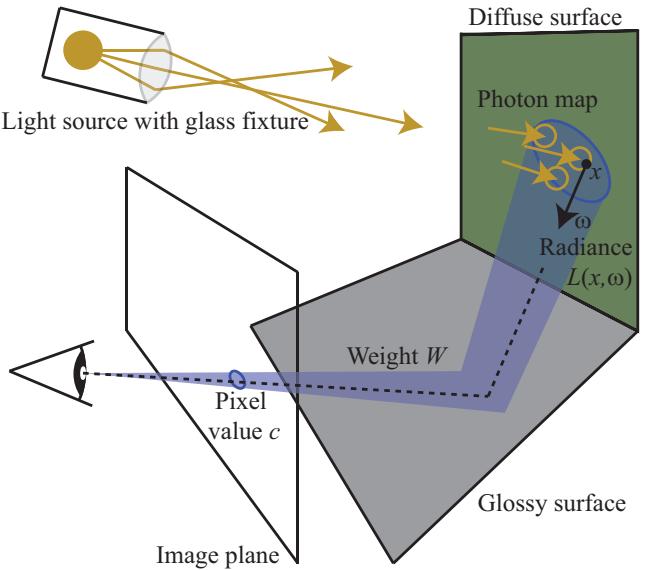


Fig. 1: This figure illustrates a typical scenario for progressive photon mapping. A light fixture enclosed by glass illuminates a scene that contains diffuse and glossy surfaces. We represent pixel values in this situation according to Equation 6.

4. PROBLEM FORMULATION

In this section we present a novel, more general derivation of progressive photon mapping (PPM) and stochastic PPM. We use a probabilistic framework that does not rely on assumptions about a specific kernel in the photon radiance estimate. It also does not rely on the maintenance of local statistics. Our goal is to compute pixel values c of the form

$$c = \int W(x, \omega) L(x, \omega) dx d\omega, \quad (6)$$

as illustrated in Figure 1. The integration is over all scene surfaces and the hemisphere of directions at each point. Here $L(x, \omega)$ is the distribution of reflected radiance over the scene surfaces. In our algorithm, we will approximate L using a photon map. In addition, W is a weighting function that describes the contribution of $L(x, \omega)$ for each surface location x and direction ω to the pixel value. This weight can include a pixel filter for antialiasing, or it can include effects such as motion blur, depth of field, and glossy reflections. In general, we will evaluate W by tracing paths from the eye.

4.1 Monte Carlo Approximation and Photon Maps

A main issue in conventional photon mapping is that there is a trade-off between the variance, or noise, and the expected error, or bias, in the radiance estimate. One can either achieve a low variance or a low expected error, but not both. The main insight of progressive photon mapping is that we can obtain a solution with, in the limit, vanishing variance and expected error by averaging the results over many photon maps.

To explain how this is possible, we first formulate the evaluation of Equation 6 as a Monte Carlo estimate. We use a photon map to obtain an approximation of the true reflected radiance. We write

the approximation as $L(x, \omega) + \epsilon$, where ϵ represents the error introduced by the radiance estimation. For brevity, we omit the arguments of the error using the shorthand $\epsilon_i = \epsilon(x, r_i)$. Using this model, our Monte Carlo estimate is

$$\bar{c}_N = \frac{1}{N} \sum_{i=1}^N \underbrace{\frac{1}{p_e(x_i, \omega_i)} W(x_i, \omega_i)}_{\text{Here, } \bar{c}_N \text{ denotes the estimate for a pixel value using } N \text{ samples, i.e., eye paths.}} (L(x_i, \omega_i) + \epsilon_i). \quad (7)$$

Here, \bar{c}_N denotes the estimate for a pixel value using N samples, i.e., eye paths. The samples (x_i, ω_i) denote position and direction of surface intersections of paths generated by ray tracing starting from the eye, and their probability density is $p_e(x_i, \omega_i)$. We denote the radiance estimation error of the i -th sample by ϵ_i .

Now the goal of our algorithm is to obtain an estimate \bar{c}_N , such that its expected value $E[\bar{c}_N]$ converges to the true pixel value c and its variance $\text{Var}[\bar{c}_N]$ approaches zero as the number of samples N increases. A crucial property of progressive photon mapping is that each sample is evaluated using a new photon distribution, i.e., a new photon map. Therefore, we interpret ϵ_i as a sample of a random variable, since the error of the radiance estimate depends on the random distribution of photons. Let us denote the average error of the photon map radiance estimate over N samples by

$$\bar{\epsilon}_N = \frac{1}{N} \sum_{i=1}^N \epsilon_i. \quad (8)$$

We show in Appendix C and D that we can achieve our goal by making sure that the variance and the expected value of the average error go to zero simultaneously, i.e.,

$$\text{as } N \rightarrow \infty, \quad \begin{aligned} \text{Var}[\bar{c}_N] &\rightarrow 0 \Rightarrow \text{Var}[\bar{c}_N] \rightarrow 0 \\ E[\bar{c}_N] &\rightarrow 0 \Rightarrow E[\bar{c}_N] \rightarrow c, \end{aligned} \quad (9)$$

where

$$\text{Var}[\bar{\epsilon}_N] = \frac{1}{N^2} \sum_{i=1}^N \text{Var}[\epsilon_i] \text{ and } E[\bar{\epsilon}_N] = \frac{1}{N} \sum_{i=1}^N E[\epsilon_i]. \quad (10)$$

We next present a strategy that allows us to fulfill these convergence conditions.

4.2 Achieving Convergence

From the last equation it is clear that the variance of the average error $\text{Var}[\bar{\epsilon}_N]$ goes to zero if the variance of each error $\text{Var}[\epsilon_i]$ stays constant. However, the expected average error $E[\bar{\epsilon}_N]$ does not vanish in this case. The main idea of progressive photon mapping is to let the variance of the radiance estimate increase by a small factor in each step, but such that in average it still vanishes. In turn, increasing the variance allows us to reduce the kernel radius, as shown in Equation 4, and this reduces the expected error, as shown in Equation 5. Based on this idea we can obtain vanishing variance and expected value of the average error simultaneously.

Variance of Average Error. Similar as in the original progressive photon mapping formulation [Hachisuka et al. 2008], in each sample we allow the variance of the error to increase by a factor

$$\frac{\text{Var}[\epsilon_{i+1}]}{\text{Var}[\epsilon_i]} = \frac{i+1}{i+\alpha}, \quad (11)$$

for some constant α with $0 < \alpha < 1$. The parameter α controls how quickly the variance is allowed to increase in each iteration.

We will see below that α effectively determines a trade-off between vanishing variance and expected value of the average error. Assuming the initial variance of the first sample is $\text{Var}[\epsilon_1]$, an explicit formula for the variance of the i -th sample (where $i > 1$) is

$$\text{Var}[\epsilon_i] = \text{Var}[\epsilon_1] \left(\prod_{k=1}^{i-1} \frac{k}{k+\alpha} \right) i. \quad (12)$$

We show in Appendix E that, using this sequence, the variance of the average error $\text{Var}[\bar{\epsilon}_N]$ is

$$\text{Var}[\bar{\epsilon}_N] = \frac{\text{Var}[\epsilon_1]}{N^2} \left(1 + \sum_{i=2}^N \left(\prod_{k=1}^{i-1} \frac{k}{k+\alpha} \right) i \right). \quad (13)$$

In addition, an asymptotic analysis of this equation shows that the variance vanishes to the order of $O(1/N^\alpha)$, i.e.,

$$\text{Var}[\bar{\epsilon}_N] = O(1/N^\alpha), \quad (14)$$

where N is the number of samples.

Expected Value of Average Error. Remember from Equation 4 that the variance is inversely proportional to the square radius. Given the sequence of variances above we therefore obtain a corresponding sequence of kernel radii,

$$\frac{r_{i+1}^2}{r_i^2} = \frac{\text{Var}[\epsilon_i]}{\text{Var}[\epsilon_{i+1}]} = \frac{i+\alpha}{i+1}. \quad (15)$$

Similar as in Equation 12, given an initial radius r_1 we get an explicit equation for radius r_i ,

$$r_i^2 = r_1^2 \left(\prod_{k=1}^{i-1} \frac{k+\alpha}{k} \right) \frac{1}{i}. \quad (16)$$

Since the expected error is proportional to the square radius (Equation 5), the expected error of the i -th sample (where $i > 1$) is

$$E[\epsilon_i] = E[\epsilon_1] \left(\prod_{k=1}^{i-1} \frac{k+\alpha}{k} \right) \frac{1}{i}, \quad (17)$$

and the expected value of the average error is

$$E[\bar{\epsilon}_N] = \frac{E[\epsilon_1]}{N} \left(1 + \sum_{i=2}^N \left(\prod_{k=1}^{i-1} \frac{k+\alpha}{k} \right) \frac{1}{i} \right). \quad (18)$$

We show in Appendix F that this vanishes to the order of $O(1/N^{1-\alpha})$, i.e.,

$$E[\bar{\epsilon}_N] = O(1/N^{1-\alpha}). \quad (19)$$

In Figure 2a we visually summarize the behavior of the variance of the error in each iteration (Equation 12) and the variance of the average error (Equation 13). The figure shows how the variance in each iteration is allowed to increase, while the variance of the average still decreases. Similarly, in Figure 2b we plot the expected error in each iteration (Equation 17) and the expected value of the average error (Equation 18). In summary, the variance and the expected value of the average error vanish as desired if $0 < \alpha < 1$, where α controls the relative rate of decrease.

We also prove in Appendix G that the radius reduction scheme proposed by Hachisuka et al. [2008] leads to the same sequence of radii as given by Equation 15, assuming that the local photon density is constant. This may be surprising, because their approach relies on

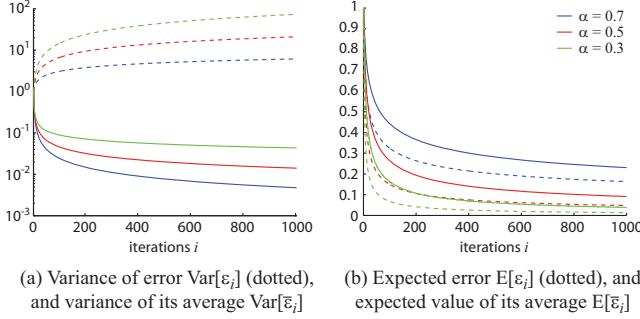


Fig. 2: This figure summarizes (a) the behavior of the variance of each iteration $\text{Var}[\epsilon_i]$ and the variance of the average over the first i iterations. Figure (b) shows the expected error of each iteration $E[\epsilon_i]$ and the expected value of the average error $E[\bar{\epsilon}_i]$ (again over the first i iterations). We plot the relative change compared to an initial variance $\text{Var}[\epsilon_1]$ and expected error $E[\epsilon_1]$.

gathering local statistics and reducing the radius based on the collected data. Our proof shows that the rate at which the radii are reduced is independent of the local photon density and gathering statistics is unnecessary.

Volumetric Progressive Photon Mapping. One of the main advantages of our derivation is that its extension to volumetric progressive photon mapping is trivial. Our model for computing pixel values from Equation 6 is equally valid for rendering participating media. In this case, radiance L is defined in a volume instead of on surfaces, and we approximate it using a volumetric photon map. In addition, the weighting function W includes attenuation along eye rays. Finally, the integration is over three spatial dimensions, instead of over surfaces.

We can use the same scheme as in Equation 12 to obtain a vanishing variance of the average error. This leads to a sequence of radii

$$\frac{r_{i+1}^3}{r_i^3} = \frac{\text{Var}[\epsilon_i]}{\text{Var}[\epsilon_{i+1}]} = \frac{i + \alpha}{i + 1} \quad (20)$$

analogous to Equation 15. Our asymptotic analysis from the appendix immediately applies to the volumetric case as well. As a result, we obtain the same convergence order of $O(1/N^\alpha)$ for the variance of the average error $\text{Var}[\bar{\epsilon}_N]$ and $O(1/N^{1-\alpha})$ for the expected average error $E[\bar{\epsilon}_N]$.

4.3 Algorithm

In this section we outline a practical algorithm based on the theory presented so far. Our algorithm simply runs a conventional photon mapping scheme with no or very small modifications in an iterative fashion. The algorithm exhibits the same convergence behavior as the original progressive photon mapping approach, even though it does not maintain any local statistics. In addition, it has the advantages that it works with arbitrary kernels, it includes volumetric photon mapping, and we can approximately predict the output pixel variance after N iterations using our asymptotic analysis.

A high level summary of our algorithm is provided in Figure 3. The main input parameters to this procedure are the desired render time,

```

1   i ← 0
2   while time is not up
3     do
4       generate photon map
5       for all pixels
6         do
7           trace path from eye until diffuse surface is hit
8           hit position and direction are  $(x_i, \omega_i)$ 
9           path contribution is  $W(x_i, \omega_i)$ 
10          path probability density is  $p_e(x_i, \omega_i)$ 
11          get current radius  $r_i$  from reference  $r_1$  and Eq. 15
12          obtain radiance estimate  $L(x_i, \omega_i) + \epsilon_i$ 
13          update pixel value, Eq. 7
14   i ← i + 1

```

Fig. 3: Pseudocode for our version of progressive photon mapping. There is no need to store local statistics. The iterations of the main loop are independent and can be performed in parallel.

the parameter α to determine the trade-off between vanishing variance and expected error, and the specification of a reference radius. The reference radius indicates the initial radius that is used for radiance estimation in the first iteration. Each iteration of the outer loop essentially performs a standard photon mapping procedure. The crucial step in our algorithm is line 11, where we compute the current radius for radiance estimation by reducing the reference by the appropriate factor from Equation 15. We discuss several possibilities to specify the reference radius below. Note that the iterations of the main loop are independent of each other and can be executed in parallel if desired. This is not possible using previous progressive photon mapping approaches, because the local statistics need to be carried over from each iteration to the next.

We have experimented with three different ways to specify the reference radii, similar as proposed by Hachisuka [2008]. The simplest option is to use a global reference radius r_1 for the whole scene. In this case, for each iteration the global radius is directly given by Equation 15. The advantage of this approach is that we can move line 11 in our algorithm to the outer loop, and there is no need to compute a local reference radius for each eye path. We can now execute the inner loop with a standard photon mapping algorithm, used as a black box, where we only need to be able to specify the appropriate radius for radiance estimation.

Alternatively, a local reference radius $r_1(x)$ can be computed per eye ray by back-projecting the pixel footprint. Here, x denotes the hit point of the eye path. Ray differentials can be used to include reflections and refractions. The advantage of this approach is that the blur introduced by radiance estimation can be kept approximately constant across the image with respect to pixel size. As a disadvantage, we need to perform the back projection calculations for each eye path.

The third option is to define the local reference radius $r_1(x)$ as the distance to the k -nearest neighbor in the photon map from location x , where radiance estimation is performed. Since we do not want to store any data, we recompute this reference radius for each radiance estimation. Again, we obtain the reduced radius for the current iteration using Equation 15. The disadvantage of this approach is that we always need to collect k photons in each iteration, while the reduced radius may only use a fraction of these photons for the actual radiance estimate. This computational overhead may lead to increased render times.

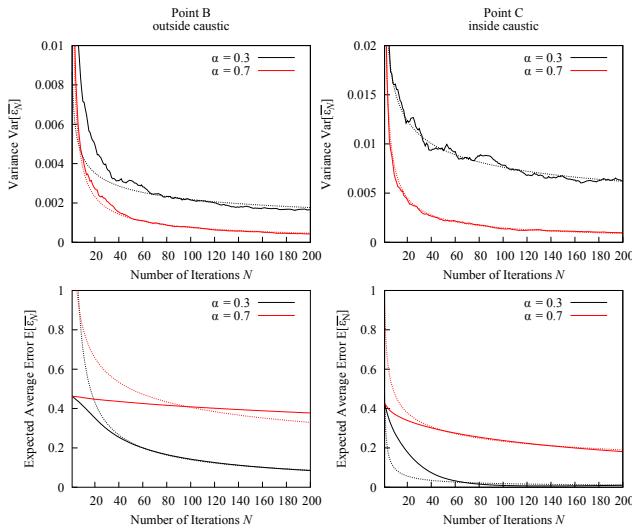


Fig. 4: Vanishing behavior of variance and expected error for pixels B and C in Figure 5. The plots in the top row show the variance $\text{Var}[\bar{\epsilon}_N]$ and the plots in the bottom row show the expected average error $E[\bar{\epsilon}_N]$ as a function of the number of iterations N . We plot the empirically estimated variance and expected average error (solid lines) and the asymptotic behavior of the variance $O(1/N^\alpha)$ and expected average error $O(1/N^{1-\alpha})$ (dotted lines). We show results for $\alpha = 0.3$ and $\alpha = 0.7$, using 100'000 photons in each iteration step. The reference radius is determined as the kNN distance in the photon maps with $k = 100$.

Note that when using local reference radii, the reference radius for a given pixel may vary from step to step during the iteration. For example, the back projection of the pixel footprint along an eye path may vary because the eye path may not be exactly the same in each iteration. To see that the variance and expected value of the average error for the pixel still converge to zero (Equation 9), we can assume that the reference radii for a pixel are bounded by some maximum and minimum value¹. Observe that the method would still converge if we *always* were to use the maximum (or always the minimum) reference radius *in each iteration*. Therefore, we also get convergence if we use any reference radius between minimum and maximum in each iteration: the variance will be smaller than if we always were to use the minimum reference, and the expected error will be smaller than if we always were to use the maximum reference.

5. RESULTS AND COMPARISONS

We first provide empirical measurements to support our analysis from the previous section. In Figure 4 we illustrate the vanishing behavior of variance $\text{Var}[\bar{\epsilon}_N]$ and expected error $E[\bar{\epsilon}_N]$ at pixels B and C in Figure 5. The top plots show the variance $\text{Var}[\bar{\epsilon}_N]$ as a function of the number of iterations N . To measure the variance empirically, we computed the result image four hundred times for each N , with different seeds for the photon maps each time. The

¹To handle pathological cases, the reference radii may need to be clamped to a user specified minimum and maximum. This may be necessary, for example, if ray differentials lead to a zero-area footprint, or if the distance to the k -nearest neighbor gets very large in completely dark areas.

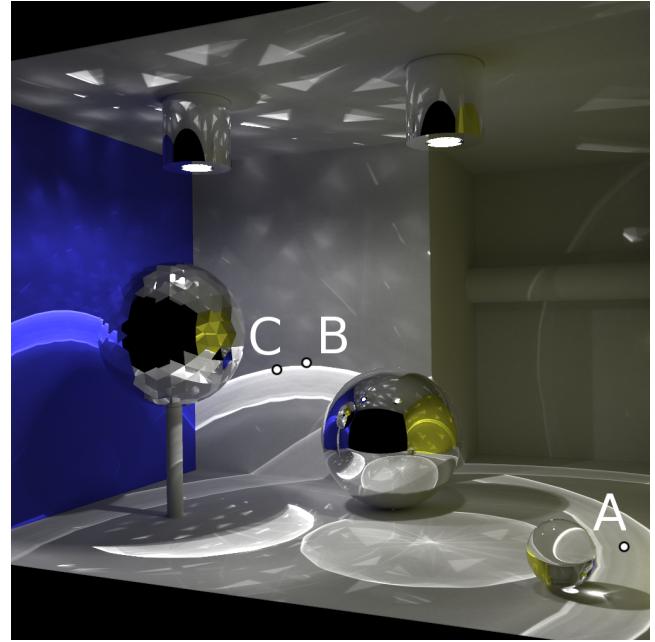


Fig. 5: This figure shows three characteristic points sampled for the statistics in Figure 4 and 6. Point A lies in a homogenous region, points B and C lie just outside and just inside the caustic.

measured variance is the sample variance of pixels B and C over these four hundred images. Note that we perform radiance estimation at the surface intersection at pixels B and C, without any further stochastic ray tracing steps. Therefore, the pixel variance is equivalent to the variance of radiance estimation. We compare the measurements to the asymptotic estimate $O(1/N^\alpha)$. The plots show that the asymptotic approximation well predicts the measured variance. The plots at the bottom visualize the convergence of the expected average error $E[\bar{\epsilon}_N]$. We empirically measure the expected average error by computing four hundred images for each N , similarly as for the variance measurement. For each N we estimate the error as the difference between the sample mean over the four hundred images and a reference image. The plots show that the measurements follow the asymptotic approximation $O(1/N^{1-\alpha})$, but only for larger N . This is not surprising considering that our analysis of the expected average error (Appendix B) is based on a second-order Taylor series expansion. The error here, however, is due to a sharp step edge that includes higher order terms. The plots indicate that for smaller N (larger radii) the higher order terms dominate the error.

In Figure 6 we compare our method with the original PPM algorithm. The plots show measurements at three characteristic points A, B, and C in Figure 5. We performed radiance estimates using a box filter and starting from a uniform global radius. We ran 200 iterations using 500'000 photons per iteration for two different alpha values $\alpha = 0.3$ and $\alpha = 0.7$. We show the squared radius r_i^2 in each iteration in the top row, and the pixel value $\bar{\epsilon}_i$ after i iterations in the bottom row.

Point A lies in a homogeneous region. There is practically no difference between the two methods, which is to be expected considering the proof in Appendix G that, for homogeneous regions, both schemes lead to the same radius reduction. Point B lies outside a

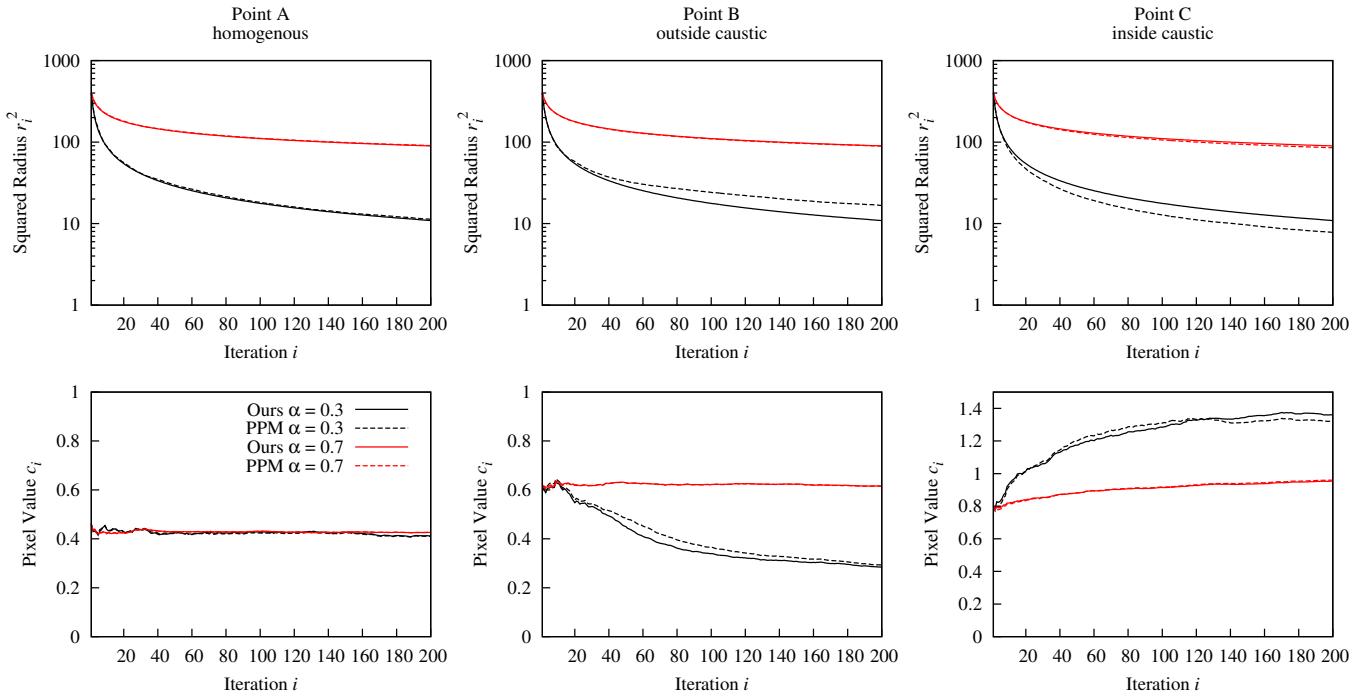


Fig. 6: We compare our method to the original PPM approach with respect to the rate of radius reduction and its effect on the resulting pixel value. The plots show measurements at three characteristic points A, B, and C in Figure 5. The top row shows the squared radius r_i^2 in each iteration, and the bottom row shows the pixel value after i iterations. Both methods shrink the radius equivalently in homogeneous regions (Point A), but the original PPM is affected by locally inhomogeneous photon densities. It shrinks the radius slower if the density is decreasing over the iterations (Point B), and faster if it is increasing (Point C).

caustic boundary, but the initial global radius covers parts of the caustic. For $\alpha = 0.3$, after about ten iterations, the radii become smaller than the distance to the caustic and the radiance begins to converge towards the correct value. For $\alpha = 0.7$, the radius reduces too slowly to leave the caustic, resulting in positive bias of the radiance even after 200 iterations. In the radius plots we observe that the original PPM shrinks the radius more slowly. This is because as the radius is reduced, a larger fraction of the box kernel lies outside the caustic and the local photon density under the kernel decreases. Point C is similar to Point B, but this time the point is inside the caustic, with the initial global radius covering the dark region. After a few iterations, the circle radius for $\alpha = 0.3$ is entirely inside the caustic, converging towards the true value. For $\alpha = 0.7$, the radius remains too big, resulting in negative bias even after 200 iterations. Here the original PPM reduces the radius faster due to the increasing photon density. In summary, our experiment confirms that the two methods are equivalent for locally homogeneous regions, but they lead to slightly different behavior if the photon density within the radiance estimation kernel is inhomogeneous. After a sufficiently large number of iterations, however, the kernels will become small enough to cover locally homogeneous areas, and we expect the same convergence to the correct, unbiased result.

Figure 7 compares rendering results of stochastic PPM and our method. We set an identical initial global reference radius to obtain a fair comparison. Our implementation uses Equation 15 to update the global radius in each step, while stochastic PPM maintains local statistics to update the radiance estimates. We evaluated

100 iterations with two million photons each. The visual results are practically identical and the difference is only in the noise.

In Figure 8 we demonstrate that with our approach we can use other kernels than a box kernel for radiance estimates. Here we used a Gaussian kernel and compare the rendering results to a box kernel. We use the exact same photon maps in both cases. The Gaussian kernel leads to slightly smoother results with similar bias.

As a proof of concept, we demonstrate progressive radiance estimation on the glossy floor surface in Figure 9. According to Equation 3 the variance of radiance estimation is proportional to the variance of the photon contributions, which is very large because of the glossy BRDF. Nonetheless the variance converges, although we obtain an acceptable result only after an impractical number of many billion photons.

Figure 10 and Figure 11 show scenes with depth of field and glossy surfaces. This demonstrates that our approach is applicable to stochastic path tracing from the eye without any modifications. In other words, our approach includes the stochastic PPM scenario [Hachisuka and Jensen 2009].

Figure 12 demonstrates the application of our approach to volumetric photon mapping. We trace two million photons per iteration. We perform several hundred iterations and trace more than a billion photons in total to obtain an image with little noise and sharp caustics.

We compare the performance of stochastic PPM with our algorithm in Table II. For three different scenes, we measured photon trac-

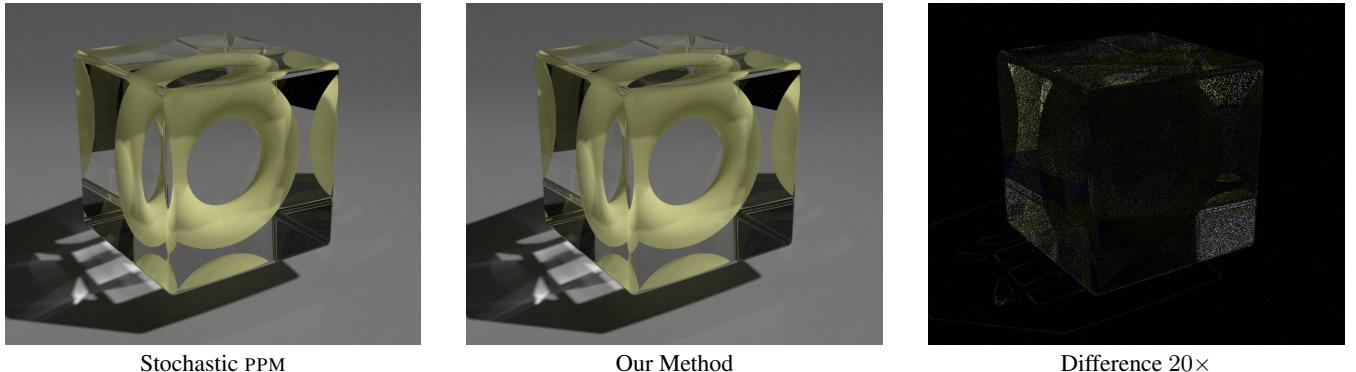


Fig. 7: Torus scene for image quality comparison between stochastic PPM and our method. We fixed the initial radius globally such that both methods have the same initial condition. The difference in the resulting images is essentially noise.

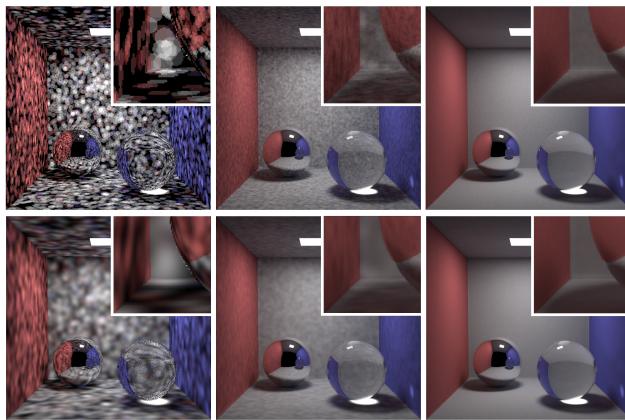


Fig. 8: We compare rendering results using a box kernel (top row) and a Gaussian kernel (bottom row). From left to right, the images were rendered using 1, 100, and 10'000 iterations. We used only 10'000 photons per iteration to emphasize the difference between the kernels. The standard deviation of the Gaussian kernel is equal to the radius of the box kernel.

ing time and eye ray tracing time including radiance estimates. The images were rendered at a resolution of 768×768 pixels using a single thread on a 2.67GHz Intel Xeon Processor using 2 million photons per iteration for 20 iterations with $\alpha = 0.7$. We tried three different strategies to initialize the reference radius, using a global reference radius, k NN-estimated radius ($k = 200$), and ray differentials. For the last two strategies, we estimate the radii in every iteration, while stochastic PPM needs to estimate the radii for the first iteration only. The cost in estimating the initial radius holds balance with the cost in maintaining local statistics such that the differences can be ignored.

As elaborated in section 4.3, one result of our analysis is that we can treat a conventional direct visualization photon mapper as a black box and extend it to PPM using a script, for example using pseudo code as in Figure 13. To implement PPM with global reference radius, we only require from the photon mapper the ability to specify the global radius to perform range queries and a seed value to generate randomized photon distributions. As a proof of concept we wrote a script for PBRT to render a series of images with



Fig. 9: We show the rendering progress using radiance estimates on a glossy surface. The images were rendered using 1, 10, 100, 1000, 10'000, and 100'000 iterations, with 1 million photons per iteration. The last image was rendered using 100 billion photons.

different radii, using Equation 15. The only change to PBRT itself was the randomization of a seed value to generate different photon distributions for every iteration. The resulting images from the iterations are then simply averaged (Figure 14). We run the script on a heterogeneous cluster with 166 nodes and various CPUs, resulting in a rendering time of 40 minutes instead of three hours on a single CPU with 8 cores. Clearly, our approach does not consume any additional memory, and it allows for parallelization using a conventional photon mapper used as a black box.

6. CONCLUSIONS

We have presented a novel formulation of progressive photon mapping based on a probabilistic perspective. Our analysis provides a conceptually simple, yet general understanding of progressive photon mapping. Using our analysis we developed a simple, memoryless algorithm that does not require the maintenance of local statistics, as is necessary in previous techniques. As a consequence, the photon mapping iterations in our algorithm are independent and



Fig. 10: The diamonds are rendered using 10 billion photons with dispersion and depth of field.



Fig. 11: The clocks are rendered using 200 million photons on a glossy surface and with depth of field.

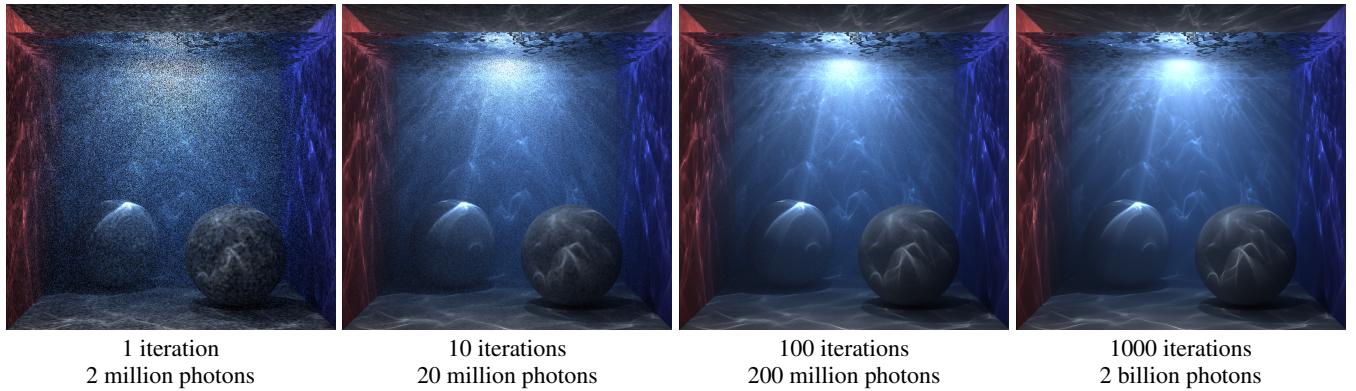


Fig. 12: We demonstrate our approach with volumetric photon mapping. We trace two million photons per iteration. We obtain a result with little noise and sharp caustics after several hundred iterations and tracing more than a billion photons in total.

```

1    $r \leftarrow r_1$ 
2   for  $i = 1 : N$ 
3       do
4            $seed \leftarrow \text{random}()$ 
5            $images(i) \leftarrow pm_{direct}(r, seed)$ 
6            $r \leftarrow r\sqrt{(i + \alpha)/(i + 1)}$ 
7    $result \leftarrow \text{average}(images, N)$ 
```

Fig. 13: Pseudo code for a script to extend a direct visualization photon mapper to PPM using a global reference radius. r is the global radius, initialized with the global reference radius r_1 .

can be run in parallel. We also obtained an asymptotic analysis that shows the trade-off between vanishing variance and expected error. In addition, our derivation holds for arbitrary radiance estimation kernels. Finally, we demonstrated a straightforward extension to participating media and volumetric photon mapping.

One direction of future investigation would be to use radiance estimates using nearest neighbor queries instead of range queries. Instead of progressively reducing the radii of the range query, we could progressively reduce the number of photons in the nearest neighbor query. Another direction would be to investigate locally varying alpha values to adaptively decrease bias in corners and boundaries. Finally, our analysis assumes that photons are distributed identically and independently. To make our approach applicable to Markov Chain Monte Carlo or adaptive importance sampling, it would be interesting to investigate how this assumption could be removed. We believe that it will be straightforward to apply our analysis to other radiance estimates beyond the usual surface and volumetric estimates, such as the beam radiance estimate. Our approach could also be useful for efficient GPU implementations because of its simplicity, and because it can be run in parallel and does not require the storage of local statistics.

Reference radius	Rendering time in seconds					
	Our method			Stochastic PPM		
	γ	Eye	Total	γ	Eye	Total
<i>Cornell Box</i>						
Global radius	179	281	459	179	288	467
k -nearest neighbour	180	268	447	180	258	437
Ray differentials	179	738	917	180	869	1048
<i>Box</i>						
Global radius	1018	132	1150	986	133	1118
k -nearest neighbour	984	117	1101	985	118	1103
Ray differentials	988	222	1210	977	238	1214
<i>Torus</i>						
Global radius	208	145	352	209	144	352
k -nearest neighbour	208	145	353	208	145	353
Ray differentials	207	244	450	208	260	467

Table II. : Performance comparison between our method and our implementation of stochastic PPM. We report timings for the photon tracing pass (γ), the eye tracing pass including radiance estimation, and the total time. The rendering times of both methods are nearly identical.

ACKNOWLEDGMENTS

We are grateful to Toshiya Hachisuka for providing the box (Figure 5 and 14), the torus (Figure 7), and the clock scenes (Figure 11). This research was supported by the Swiss National Science Foundation under grant number 200021-127166. Finally, we would like to thank the anonymous reviewers for their insightful comments and suggestions.

APPENDIX

A. VARIANCE OF RADIANCE ESTIMATE

The error of the radiance estimate using a kernel k_r of radius r is represented by

$$\epsilon(x, r) = \frac{1}{M} \sum_{j=1}^M k_r(x_j - x) \gamma_j - L(x, \omega).$$

Note that the dependency of the radiance estimate on the outgoing direction ω is hidden in the photon values γ_j , which include multiplication with the BRDF.

We now analyze the variance $\text{Var}[\epsilon(x, r)]$ of the error of the radiance estimate $\epsilon(x, r)$. We interpret photon positions x_j as independent, identically distributed (i.i.d.) samples of a random variable X with a certain probability density $p_l(x)$. Likewise, we treat the photon contributions γ_j as samples of a random variable γ with some probability distribution. We have

$$\begin{aligned} \text{Var}[\epsilon(x, r)] &= \text{Var} \left[\frac{1}{M} \sum_{j=1}^M k_r(X - x) \gamma_j - L(x, \omega) \right] \\ &= \frac{1}{M} \text{Var}[k_r(X - x)] (\text{Var}[\gamma] + \text{E}[\gamma]^2) \\ &\quad + \frac{1}{M} \text{Var}[\gamma] \text{E}[k_r(X - x)]^2, \end{aligned}$$

where we assumed that the random variables X and γ are independent.

Let us assume that the kernel k_r is compactly supported in an area Ω and that the probability density of the photons is constant over Ω . We call this density $p_l(x)$. We note that the expected value $\text{E}[k_r(X - x)]$ is $p_l(x)$. We also express the kernel k_r in terms of a canonic kernel k with unit scale and unit integral, i.e., $k_r(\xi) = 1/r^2 k(\xi/r)$ and obtain

$$\int_{\Omega} k_r(\xi - x)^2 d\xi = \frac{1}{r^2} \int_{\mathbb{R}^2} k(\psi)^2 d\psi,$$

where we used a variable substitution $\psi = (\xi - x)/r$, $d\xi = r^2 d\psi$. The factor r^2 appears because this is a two-dimensional integral. The variance $\text{Var}[k_r(X - x)]$ can now be written as

$$\text{Var}[k_r(X - x)] = \frac{p_l(x)}{r^2} \int_{\mathbb{R}^2} k(\psi)^2 d\psi - p_l^2(x).$$

Substituting this into the earlier expression for the variance of the error of the radiance estimate, $\text{Var}[\epsilon(x, r)]$, we obtain

$$\begin{aligned} \text{Var}[\epsilon(x, r)] &= \frac{1}{M} \left(\frac{p_l(x)}{r^2} \int_{\mathbb{R}^2} k(\psi)^2 d\psi - p_l^2(x) \right) (\text{Var}[\gamma] + \text{E}[\gamma]^2) \\ &\quad + \frac{1}{M} \text{Var}[\gamma] p_l^2(x) \\ &\approx (\text{Var}[\gamma] + \text{E}[\gamma]^2) \frac{p_l(x)}{Mr^2} \int_{\mathbb{R}^2} k(\psi)^2 d\psi. \end{aligned}$$

In the last step we make the assumption that the kernel covers only a small part of the whole scene, hence $\int_{\mathbb{R}^2} k(\psi)^2 d\psi \gg p_l^2(x)$.

B. EXPECTED ERROR OF RADIANCE ESTIMATE

The expected error of the radiance estimate is defined as

$$\begin{aligned} \text{E}[\epsilon(x, r)] &= \text{E} \left[\frac{1}{M} \sum_{j=1}^M k_r(X - x) \gamma_j - L(x, \omega) \right] \\ &= \text{E}[\gamma] \text{E}[k_r(X - x)] - L(x, \omega). \end{aligned}$$

Again, the photon positions are interpreted as i.i.d. samples of a random variable X with a certain density $p_l(x)$ and the photon contributions are interpreted as samples of a random variable γ . The expected value $\text{E}[k_r(X - x)]$ is defined as the integral

$$\text{E}[k_r(X - x)] = \frac{1}{r^2} \int_{\mathbb{R}^2} k((\xi - x)/r) p_l(\xi) d\xi.$$

To obtain an estimate of this, we use a Taylor expansion of the photon density $p_l(\xi)$ around x ,

$$p_l(\xi) = p_l(x) + (\xi - x) \nabla p_l(x) + O(\|\xi - x\|^2),$$

and substitute it into the integral. Similar approaches are used to analyse the properties of density estimation in statistics [Silverman 1986]. In addition, we make a change of integration variables $\psi = (\xi - x)/r$, $d\xi = r^2 d\psi$. The factor here is r^2 because the kernel is two-dimensional. Note that $O(\|\xi - x\|^2) = r^2 O(\|\psi\|^2)$. Exploiting the fact that k has unit integral and, in most practical cases, a vanishing first moment, i.e., $\int_{\mathbb{R}^2} \psi k(\psi) d\psi = 0$, we obtain

$$\text{E}[k_r(X - x)] = p_l(x) + r^2 \int_{\mathbb{R}^2} k(\psi) O(\|\psi\|^2) d\psi.$$

We note that the expected value of a radiance estimate with infinitely small kernel corresponds to the exact radiance, hence we can express $L(x, \omega)$ as

$$L(x, \omega) = E[\gamma] E[\delta(X - x)] = E[\gamma] p_l(x),$$

where δ is the Dirac delta function. Finally, this means

$$\begin{aligned} E[\epsilon(x, r)] &= E[\gamma] E[k_r(X - x)] - L(x, \omega) \\ &= r^2 E[\gamma] \int_{\mathbb{R}^2} k(\psi) O(\|\psi^2\|) d\psi \\ &= r^2 E[\gamma] \tau, \end{aligned}$$

for some constant τ . Unfortunately it is difficult in practice to estimate the τ because it depends on the higher order derivatives of the photon density distribution $p_l(x)$.

C. VARIANCE OF THE PIXEL ESTIMATE

Here we derive the variance of the pixel estimate \bar{c}_N after N photon mapping steps. We show that, as the number N of samples goes to infinity, the variance of \bar{c}_N vanishes, if the variance of the average error $\text{Var}[\bar{\epsilon}_N]$ of the radiance estimates vanishes. Here the samples (x_i, ω_i) are hit positions of eye paths, not photons. But similar as before, the samples are i.i.d. with a single density $p_e(x, \omega)$. Hence, they are represented by a single random variable when computing variance or expected value. To simplify notation even further, we omit this random variable as argument of the functions L , W , and p_e . We have

$$\begin{aligned} \text{Var}[\bar{c}_N] &= \text{Var}\left[\frac{1}{N} \sum_{i=1}^N \frac{W}{p_e} (L + \epsilon_i)\right] \\ &= \frac{1}{N^2} \sum_{i=1}^N \text{Var}\left[\frac{W}{p_e} L\right] + \frac{1}{N^2} \sum_{i=1}^N \text{Var}\left[\frac{W}{p_e} \epsilon_i\right] \\ &= \frac{1}{N} \text{Var}\left[\frac{W}{p_e} L\right] + \frac{1}{N^2} \sum_{i=1}^N \text{Var}\left[\frac{W}{p_e} \epsilon_i\right]. \end{aligned}$$

Note that the errors of the radiance estimate in each step ϵ_i are random variables that are *not* identically distributed in progressive photon mapping. The first sum in the last line is the usual variance of the Monte Carlo estimate, which converges to zero with $1/N$. Assuming the random variables W/p_e and ϵ_i are independent, we rewrite the second sum as

$$\begin{aligned} \frac{1}{N^2} \sum_{i=1}^N \text{Var}\left[\frac{W}{p_e} \epsilon_i\right] &= \text{Var}\left[\frac{W}{p_e}\right] \frac{1}{N^2} \sum_{i=1}^N \text{Var}[\epsilon_i] \\ &+ E\left[\frac{W}{p_e}\right]^2 \frac{1}{N^2} \sum_{i=1}^N \text{Var}[\epsilon_i] + \text{Var}\left[\frac{W}{p_e}\right] \frac{1}{N^2} \sum_{i=1}^N E[\epsilon_i]^2. \end{aligned}$$

Here, the last sum vanishes with $N \rightarrow \infty$ because $E[\epsilon_i] \leq E[\epsilon_1]$, which follows from Equation 17. The other parts vanish if

$$\frac{1}{N^2} \sum_{i=1}^N \text{Var}[\epsilon_i] = \text{Var}[\bar{\epsilon}_N],$$

vanishes, as we claimed in the beginning.

D. EXPECTED VALUE OF THE PIXEL ESTIMATE

Here we derive the expected value of the pixel estimate \bar{c}_N . We show that as the number of samples N goes to infinity, the expected value of \bar{c}_N converges to the exact value c , if the expected value of the average error $E[\bar{\epsilon}_N]$ of the radiance estimates vanishes. The expected value of the pixel estimate is

$$\begin{aligned} E[\bar{c}_N] &= E\left[\frac{1}{N} \sum_{i=1}^N \frac{1}{p_e} W(L + \epsilon_i)\right] \\ &= \frac{1}{N} \sum_{i=1}^N E\left[\frac{1}{p_e} WL\right] + \frac{1}{N} \sum_{i=1}^N E\left[\frac{1}{p_e} W\right] E[\epsilon_i] \\ &= c + E\left[\frac{1}{p_e} W\right] \frac{1}{N} \sum_{i=1}^N E[\epsilon_i] \\ &= c + E\left[\frac{1}{p_e} W\right] E[\bar{\epsilon}_N] \end{aligned}$$

where we used the fact that the expected value of the Monte Carlo estimate corresponds to the exact pixel value c . We conclude that if the expected value of the average error $E[\bar{\epsilon}_N]$ vanishes, then the expected pixel value is the exact value, $E[\bar{c}_N] = c$.

E. VARIANCE OF THE AVERAGE ERROR

The variance of the average error over the PPM iterations is

$$\text{Var}[\bar{\epsilon}_N] = \text{Var}\left[\frac{1}{N} \sum_{i=1}^N \epsilon_i\right] = \frac{1}{N^2} \sum_{i=1}^N \text{Var}[\epsilon_i].$$

Now we show that $\text{Var}[\bar{\epsilon}_N]$ converges even if we increase the variance $\text{Var}[\epsilon_i]$ in each step. We allow the variance to increase by a factor

$$\text{Var}[\epsilon_{i+1}] = \frac{i+1}{i+\alpha} \text{Var}[\epsilon_i]$$

in each step. Starting with a variance $\text{Var}[\epsilon_1]$, an equivalent explicit formula for $i > 1$ is

$$\text{Var}[\epsilon_i] = \text{Var}[\epsilon_1] \left(\prod_{k=1}^{i-1} \frac{k}{k+\alpha} \right) i.$$

Therefore, the variance of the average error is

$$\text{Var}[\bar{\epsilon}_N] = \frac{\text{Var}[\epsilon_1]}{N^2} \left(1 + \sum_{i=2}^N \left(\prod_{k=1}^{i-1} \frac{k}{k+\alpha} \right) i \right).$$

To analyze this equation asymptotically, we note that for large i we can approximate the product formula as

$$\begin{aligned} \left(\prod_{k=1}^{i-1} \frac{k}{k+\alpha} \right) i &= \frac{i\alpha\Gamma(\alpha)\Gamma(i)}{\Gamma(\alpha+i)} \\ &= i\alpha B(\alpha, i) \\ &= \Theta(i^{1-\alpha}), \end{aligned}$$

where we use the Gamma and Beta functions Γ and B , respectively, and Θ is asymptotic notation. The asymptotic approximation is based on Stirling's formula [Abramowitz and Stegun 1964].

Finally we get

$$\begin{aligned}\text{Var}[\bar{\epsilon}_N] &= \frac{\text{Var}[\epsilon_1]}{N^2} \left(1 + \sum_{i=2}^N \left(\prod_{k=1}^{i-1} \frac{k}{k+\alpha} \right) i \right) \\ &= \frac{\text{Var}[\epsilon_1]}{N^2} \left(1 + \sum_{i=2}^N \Theta(i^{1-\alpha}) \right) \\ &= \frac{\text{Var}[\epsilon_1]}{N^2} (1 + NO(N^{1-\alpha})) \\ &= O(1/N^\alpha).\end{aligned}$$

This confirms the intuition that for $\alpha = 1$, convergence is proportional to $1/N$. In addition, it shows that there is no convergence for $\alpha = 0$.

F. EXPECTED AVERAGE ERROR

Recall the expected average error over the PPM iterations from Equation 18,

$$\begin{aligned}E[\bar{\epsilon}_N] &= \frac{E[\epsilon_1]}{N} \left(1 + \sum_{i=2}^N \left(\prod_{k=1}^{i-1} \frac{k+\alpha}{k} \right) \frac{1}{i} \right) \\ &= \frac{E[\epsilon_1]}{N} \left(1 + \sum_{i=2}^N \frac{1}{B(\alpha, i)\alpha i} \right) \\ &= \frac{E[\epsilon_1]}{N} \sum_{i=1}^N \frac{1}{B(\alpha, i)\alpha i}\end{aligned}$$

We show in Appendix H that this series of Beta functions can be expressed in closed form, and it follows that

$$\sum_{k=1}^N \frac{1}{B(\alpha, k)k} = \frac{1}{B(\alpha, N)\alpha} + \frac{1}{B(\alpha, N)N} - 1.$$

Substituting this expression in the above equation yields

$$\begin{aligned}E[\bar{\epsilon}_N] &= \frac{E[\epsilon_1]}{N} \frac{1}{\alpha} \left(\frac{1}{B(\alpha, N)\alpha} + \frac{1}{B(\alpha, N)N} - 1 \right) \\ &= \frac{E[\epsilon_1]}{N} \left(\frac{\alpha + N}{B(\alpha, N)\alpha^2 N} - \frac{1}{\alpha} \right) \\ &= \frac{E[\epsilon_1]}{N} \left(\Theta(N^{\alpha-1}) \frac{\alpha + N}{\alpha} - \frac{1}{\alpha} \right) \\ &= O(N^{\alpha-1}).\end{aligned}$$

G. EQUIVALENCE WITH ORIGINAL PPM

We prove that the radius reduction scheme by Hachisuka et al. [2008] is the same as our approach in Equation 15. The local radius reduction factor proposed by Hachisuka et al. is given in Equation 9 of their paper,

$$\frac{r_{i+1}^2}{r_i^2} = \frac{N_i + \alpha M_i}{N_i + M_i},$$

where M_i is the number of photons collected in the current iteration, and N_i is an accumulated statistics (note that we abuse notation here to be consistent with the Hachisuka et al.; in the rest of our

paper N is the number of iterations). We derive an explicit formula for N_i from Equation 7 in Hachisuka et al.,

$$N_i = N_{i-1} + \alpha M_{i-1} = N_1 + \alpha \sum_{k=1}^{i-1} M_k,$$

and substitute it above to obtain

$$\frac{r_{i+1}^2}{r_i^2} = \frac{N_1 + \alpha \sum_{k=1}^{i-1} M_k + \alpha M_i}{N_1 + \alpha \sum_{k=1}^{i-1} M_k + M_i}.$$

In the first iteration the number of photons found within r_1 is N_1 . For the next iteration, prior to radius reduction, we find M_1 photons using the *same* radius r_1 . Therefore $N_1 = M_1$ and we get

$$\frac{r_{i+1}^2}{r_i^2} = \frac{M_1 + \alpha \sum_{k=1}^{i-1} M_k + \alpha M_i}{M_1 + \alpha \sum_{k=1}^{i-1} M_k + M_i}.$$

Assuming the photon density $d = M_i / (\pi r_i^2)$ is locally constant, we substitute $M_i = d\pi r_i^2$ and obtain a recursive formula for r_i^2 ,

$$\frac{r_{i+1}^2}{r_i^2} = \frac{r_1^2 + \alpha \sum_{k=1}^{i-1} r_k^2 + \alpha r_i^2}{r_1^2 + \alpha \sum_{k=1}^{i-1} r_k^2 + r_i^2}.$$

To guess an explicit expression for the sequence of radii, we develop the first three iteration steps:

$$\begin{aligned}r_2^2 &= r_1^2 \frac{\alpha + 1}{2} \\ r_3^2 &= r_1^2 \frac{(\alpha + 1)(\alpha + 2)}{6} \\ r_4^2 &= r_1^2 \frac{(\alpha + 1)(\alpha + 2)(\alpha + 3)}{24}.\end{aligned}$$

We write the above using a product formula and observe that we obtain a Pochhammer polynomial. This allows us to express our formula using the Euler Gamma and Beta functions,

$$r_i^2 = r_1^2 \frac{\prod_{k=1}^{i-1} (\alpha + k)}{i!} = r_1^2 \frac{\Gamma(\alpha + i)}{\Gamma(\alpha + 1)\Gamma(i + 1)} = \frac{r_1^2}{B(\alpha, i)\alpha i}.$$

In addition, our guess leads to a simple recursive formula for the sequence of radii,

$$\frac{r_{i+1}^2}{r_i^2} = \frac{i + \alpha}{i + 1}.$$

We now prove our guess using induction. It is obvious that it holds for the anchor. Next we show it also holds for the induction step $r_i^2 \rightarrow r_{i+1}^2$. We restate our claim,

$$\frac{r_{i+1}^2}{r_i^2} = \frac{r_1^2 + \alpha \sum_{k=1}^{i-1} r_k^2 + \alpha r_i^2}{r_1^2 + \alpha \sum_{k=1}^{i-1} r_k^2 + r_i^2} = \frac{i + \alpha}{i + 1},$$

and multiply out the second identity to obtain

$$r_i^2 = \frac{r_1^2 + \alpha \sum_{k=1}^{i-1} r_k^2}{i}.$$

We replace r_i^2 and r_k^2 by the explicit formula, i.e., our induction hypothesis, and get a recursive definition of the Beta function,

$$\frac{r_1^2}{B(\alpha, i)\alpha i} = \frac{1}{i} \left(r_1^2 + \alpha \sum_{k=1}^{i-1} \frac{r_1^2}{B(\alpha, k)\alpha k} \right),$$

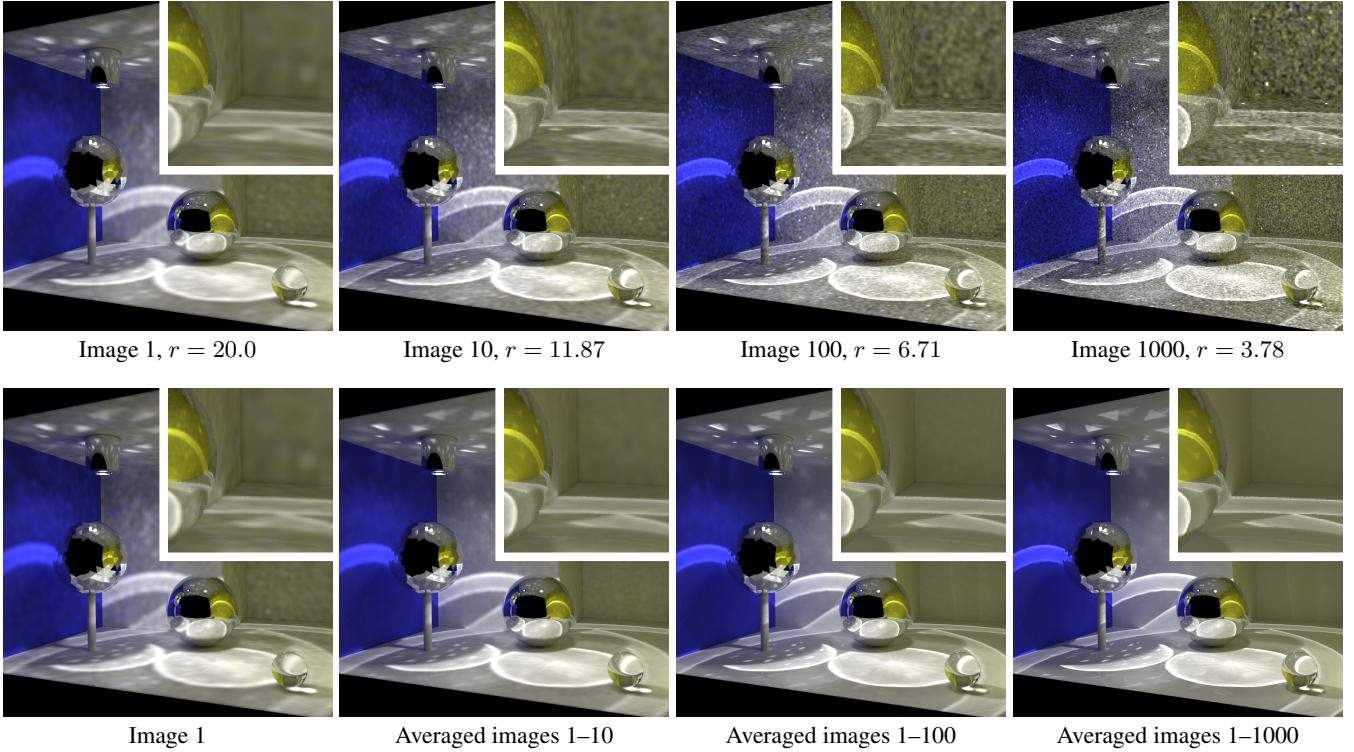


Fig. 14: Our formulation of PPM can be used to extend a conventional direct visualization photon mapper to PPM using a simple script (Figure 13). We rendered the box scene using PBRT on a cluster. The script submitted 1000 independent render jobs with radii obtained from the radius sequence defined by Equation 15 using $\alpha = 0.5$. Every image was rendered using three million stored photons. The top row shows the rendered images for iteration number 1, 10, 100, and 1000. The bottom row shows the averaged results over the corresponding sequence.

which is equivalent to

$$\frac{1}{B(\alpha, i)\alpha} = 1 + \sum_{k=1}^{i-1} \frac{1}{B(\alpha, k)k}.$$

We prove this identity in Appendix H.

H. RECURSIVE FORMULA OF BETA FUNCTION

We use induction to prove the identity

$$\frac{1}{B(\alpha, i)\alpha} = 1 + \sum_{k=1}^{i-1} \frac{1}{B(\alpha, k)k}.$$

Splitting up the sum and substituting the induction hypothesis, we get

$$\begin{aligned} \frac{1}{B(\alpha, i+1)\alpha} &= 1 + \sum_{k=1}^{i-1} \frac{1}{B(\alpha, k)k} + \frac{1}{B(\alpha, i)i} \\ &= \frac{1}{B(\alpha, i)\alpha} + \frac{1}{B(\alpha, i)i}, \end{aligned}$$

which is equivalent to a well known functional equation for the Beta function,

$$B(\alpha, i+1) = B(\alpha, i) \frac{i}{i+\alpha}.$$

REFERENCES

- ABRAMOWITZ, M. AND STEGUN, I. A. 1964. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, ninth Dover printing, tenth GPO printing ed. Dover, New York.
- CAMMARANO, M. AND JENSEN, H. W. 2002. Time dependent photon mapping. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering*. 135–144.
- CLINE, D., TALBOT, J., AND EGBERT, P. 2005. Energy redistribution path tracing. *ACM Trans. Graph.* 24, 3.
- HACHISUKA, T., JAROSZ, W., AND JENSEN, H. W. 2010. A progressive error estimation framework for photon density estimation. *ACM Trans. Graph.* 29, 6.
- HACHISUKA, T. AND JENSEN, H. W. 2009. Stochastic progressive photon mapping. *ACM Trans. Graph.* 28, 5.
- HACHISUKA, T., OGAKI, S., AND JENSEN, H. W. 2008. Progressive photon mapping. *ACM Trans. Graph.* 27, 5.
- HERZOG, R., HAVRAN, V., KINUWAKI, S., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2007. Global illumination using photon ray splatting. *26*, 3, 503–513.
- HERZOG, R. AND SEIDEL, H.-P. 2007. Lighting details preserving photon density estimation. In *The 15th Pacific Conference on Computer Graphics and Application*. 407–410.
- JAROSZ, W., ZWICKER, M., AND JENSEN, H. W. 2008. The Beam Radiance Estimate for Volumetric Photon Mapping. *Computer Graphics Forum (Proc. Eurographics EG'08)* 27, 2, 557–566.

- JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. AK Peters. ISBN: 1568811470.
- JENSEN, H. W. AND CHRISTENSEN, P. H. 1998. Efficient simulation of light transport in scenes with participating media using photon maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 311–320.
- KAJIYA, J. T. 1986. The rendering equation. In *Computer Graphics (Proc. SIGGRAPH 86)*. 143–150.
- LAFORTUNE, E. P. AND WILLEMS, Y. D. 1993. Bi-directional path tracing. In *Compugraphics '93*. 145–153.
- PAULY, M., KOLLIG, T., AND KELLER, A. 2000. Metropolis light transport for participating media. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*. 11–22.
- PHARR, M. AND HUMPHREYS, G. 2004. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann.
- SCHREGLE, R. 2003. Bias compensation for photon maps. *Computer Graphics Forum* 22, 4, 729–742.
- SILVERMAN, B. W. 1986. *Density estimation in statistics and data analysis*. Chapman & Hall/CRC.
- SPENCER, B. AND JONES, M. 2009. Into the blue: Better caustics through photon relaxation. *Computer Graphics Forum* 28, 2, 319–328.
- VEACH, E. 1998. Robust monte carlo methods for light transport. Ph.D. thesis, Stanford University.
- VEACH, E. AND GUIBAS, L. J. 1995. Optimally combining sampling techniques for monte carlo rendering. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 419–428.
- VEACH, E. AND GUIBAS, L. J. 1997. Metropolis light transport. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*. 65–76.