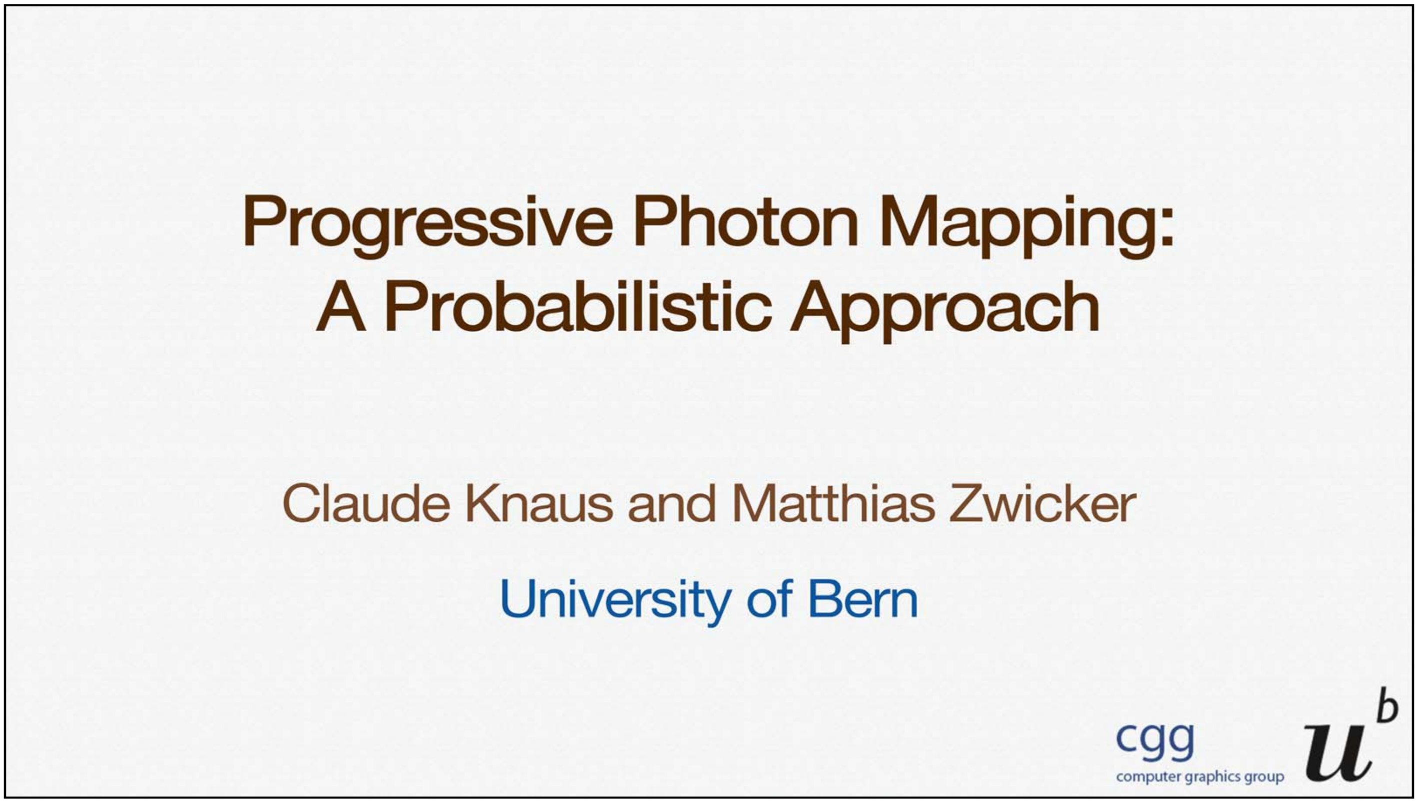
SIGGRAPH2011



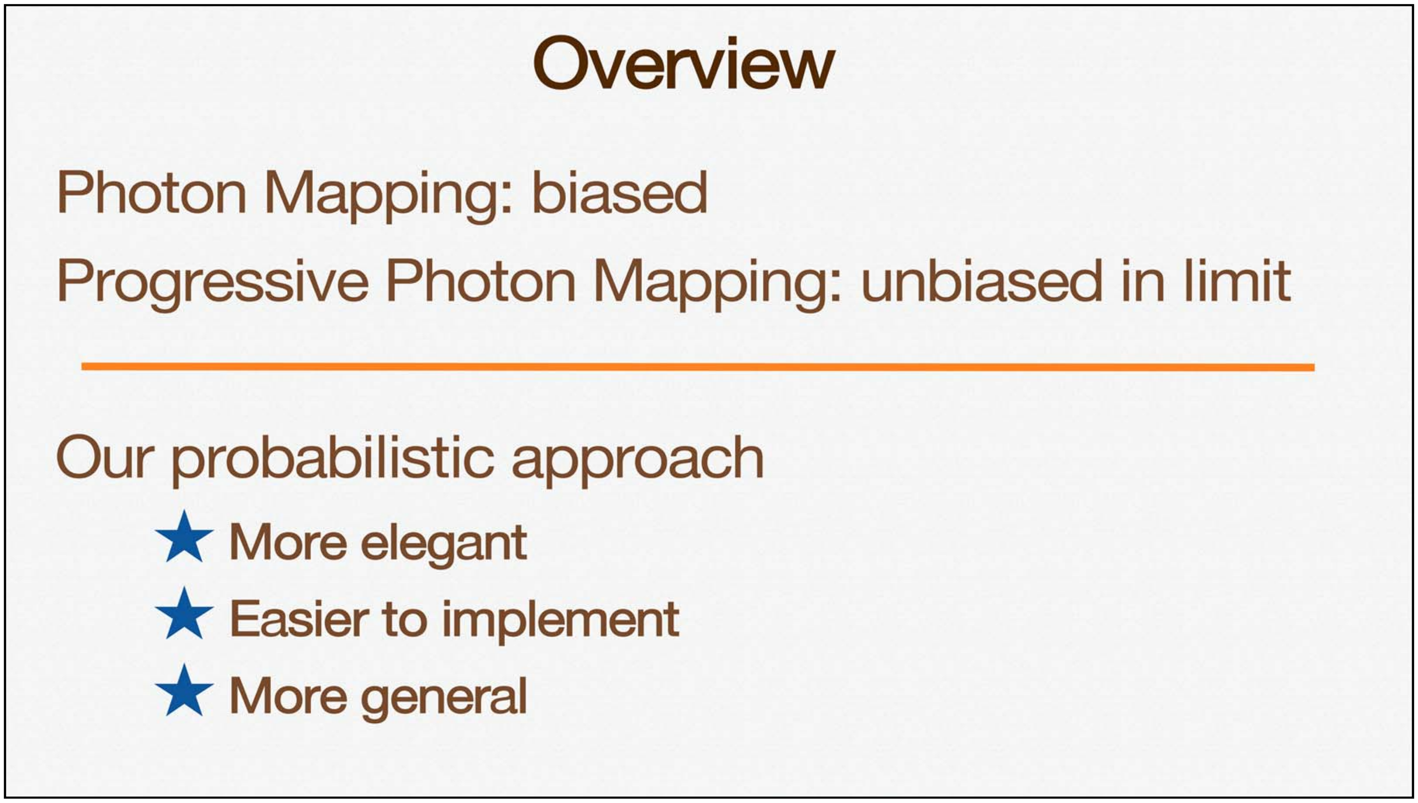
Progressive Photon Mapping: A Probabilistic Approach

Claude Knaus and Matthias Zwicker

University of Bern

cgg
computer graphics group

Thank you for the introduction.



Overview

Photon Mapping: biased

Progressive Photon Mapping: unbiased in limit

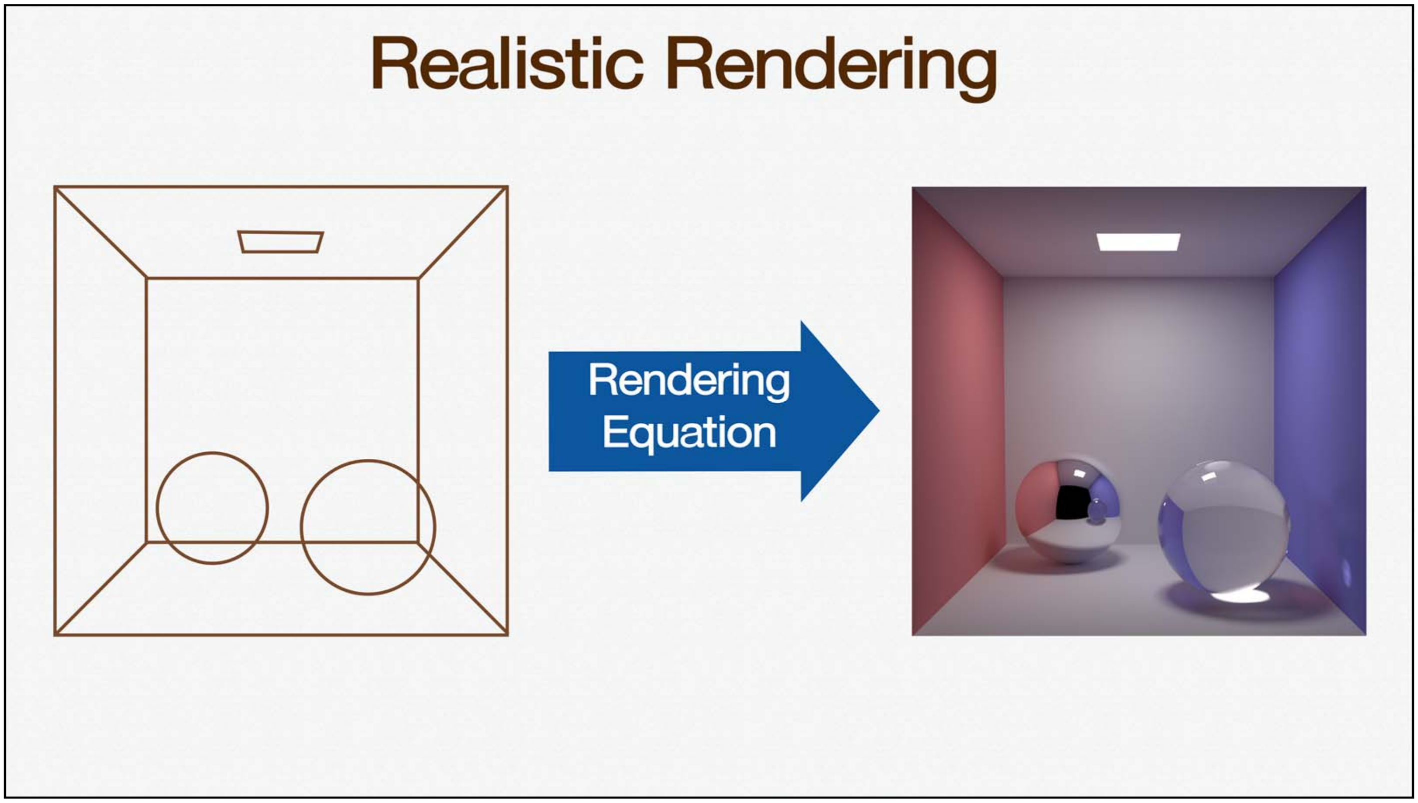
Our probabilistic approach

★ More elegant

★ Easier to implement

★ More general

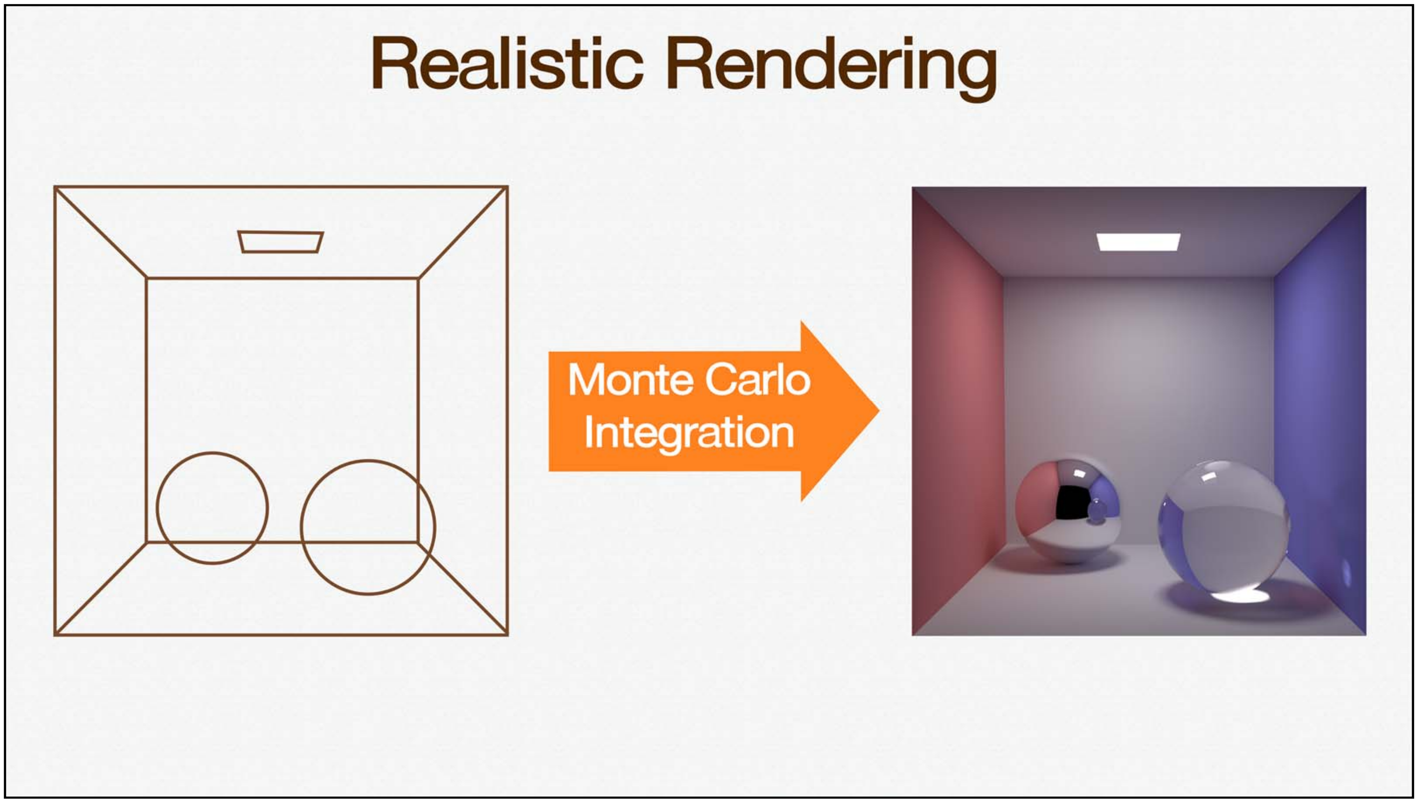
This talk consists of two parts. In the first part, we will review photon mapping and its problem of being biased, which means that there is a consistent error in the rendered image. This problem has been solved recently by progressive photon mapping, which is unbiased in the limit. In the second part, we propose an alternative, a probabilistic approach to progressive photon mapping, which is more elegant, easier to implement, and also more general.



Realistic Rendering

Rendering Equation

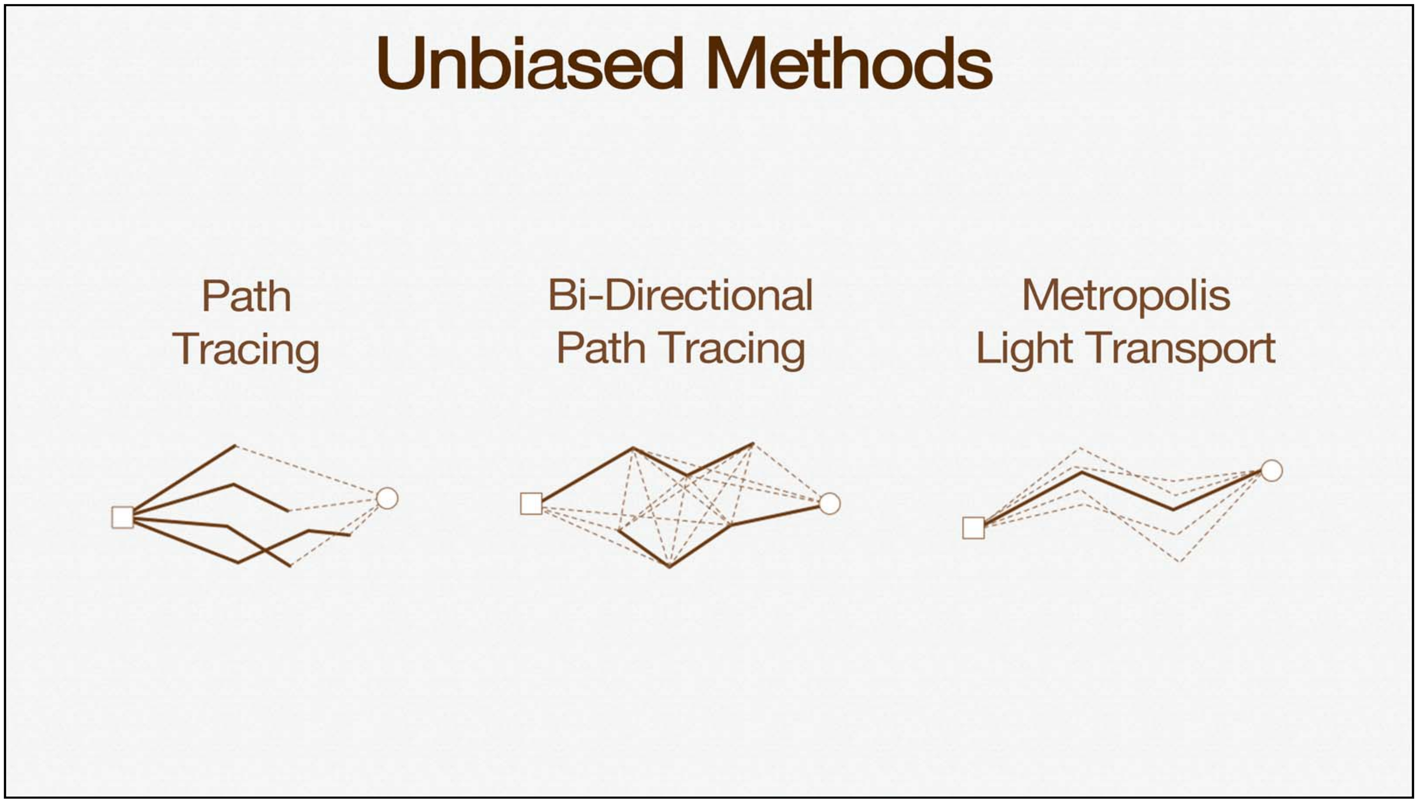
Realistic rendering is based on Kajiya’s rendering equation. The rendering equation involves an integral, which is usually numerically solved using Monte Carlo integration.



Realistic Rendering

Monte Carlo Integration

Methods using Monte Carlo integration sum over randomly sampled light paths. The most commonly used methods exploiting Monte Carlo integration to solve the rendering equation are



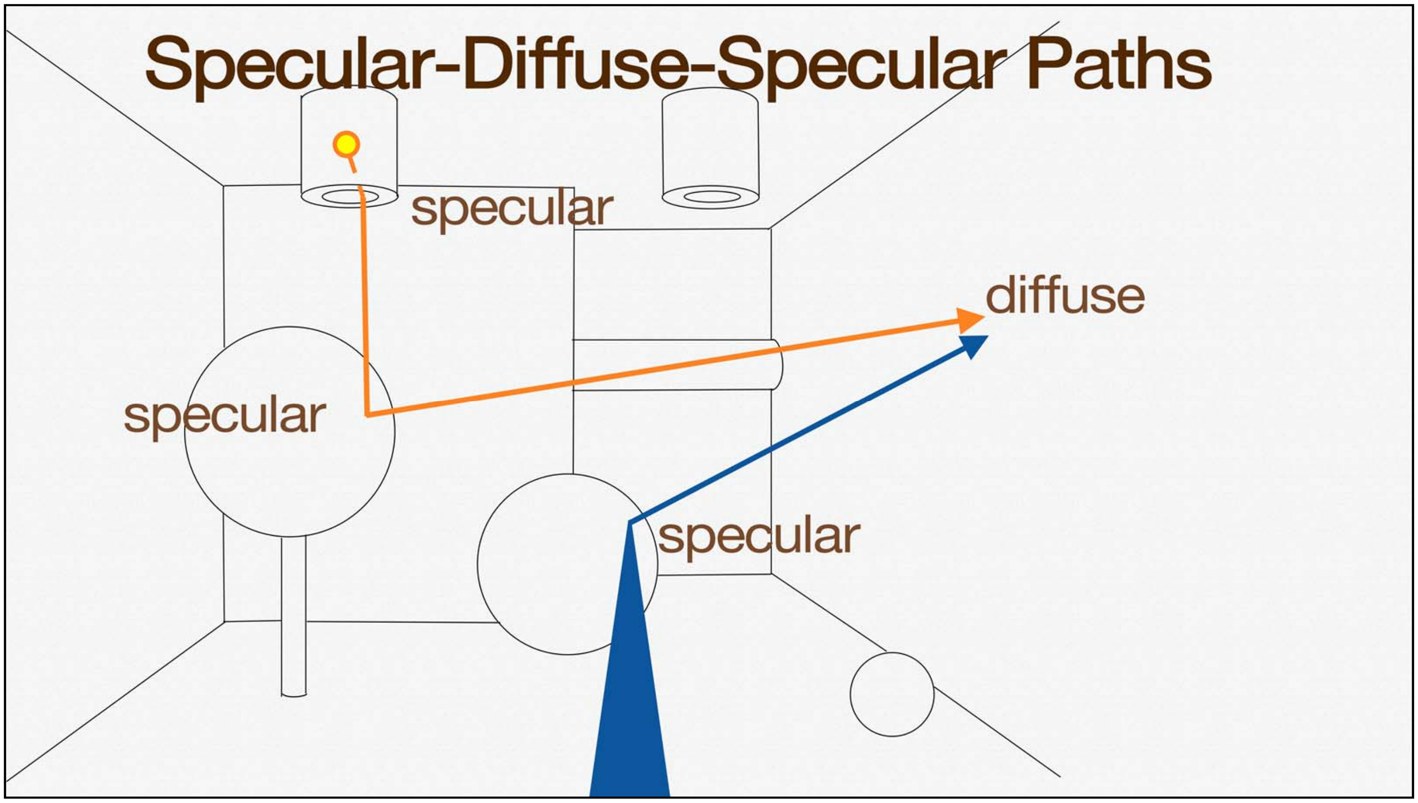
Unbiased Methods

Path Tracing

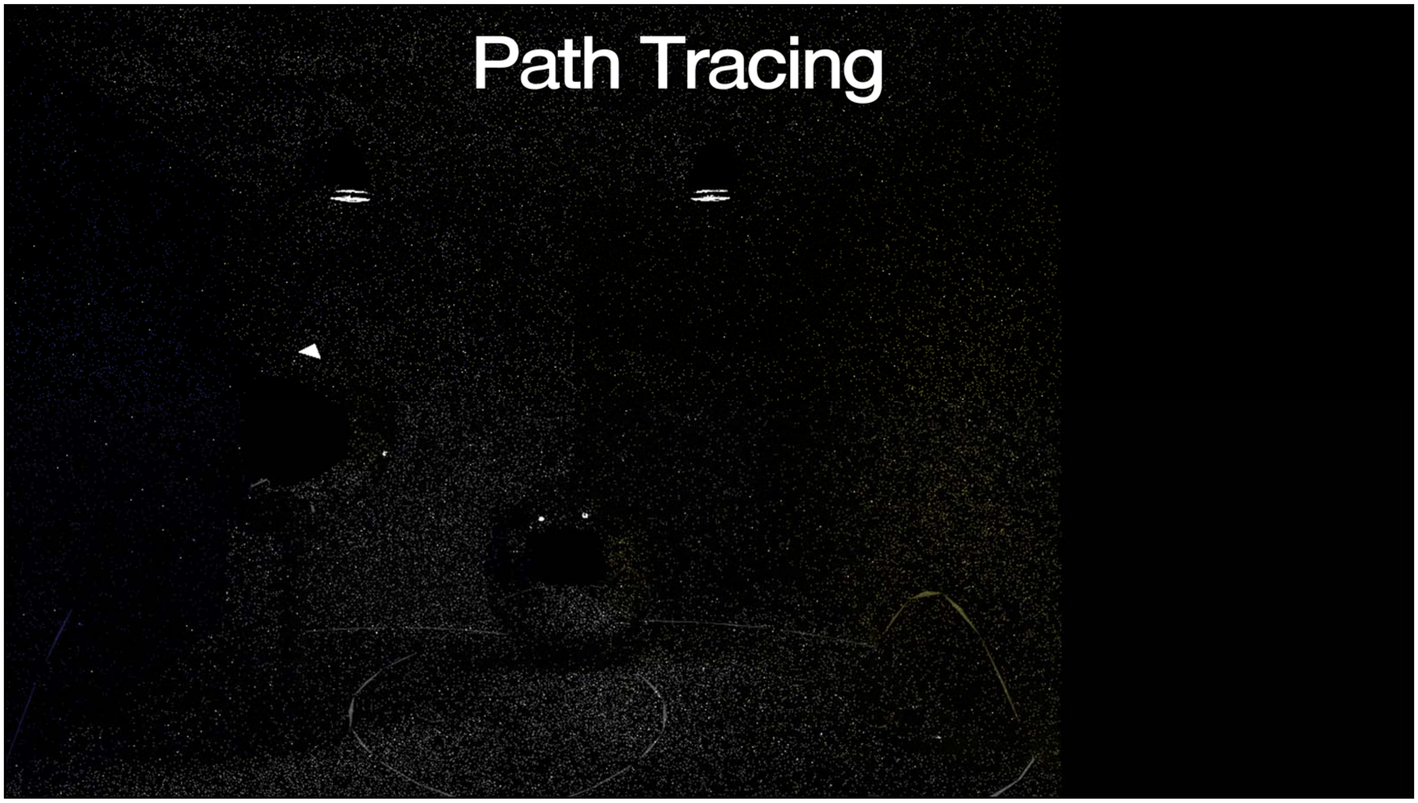
Bi-Directional Path Tracing

Metropolis Light Transport

path tracing, bi-directional path tracing, and metropolis light transport. These are unbiased methods.

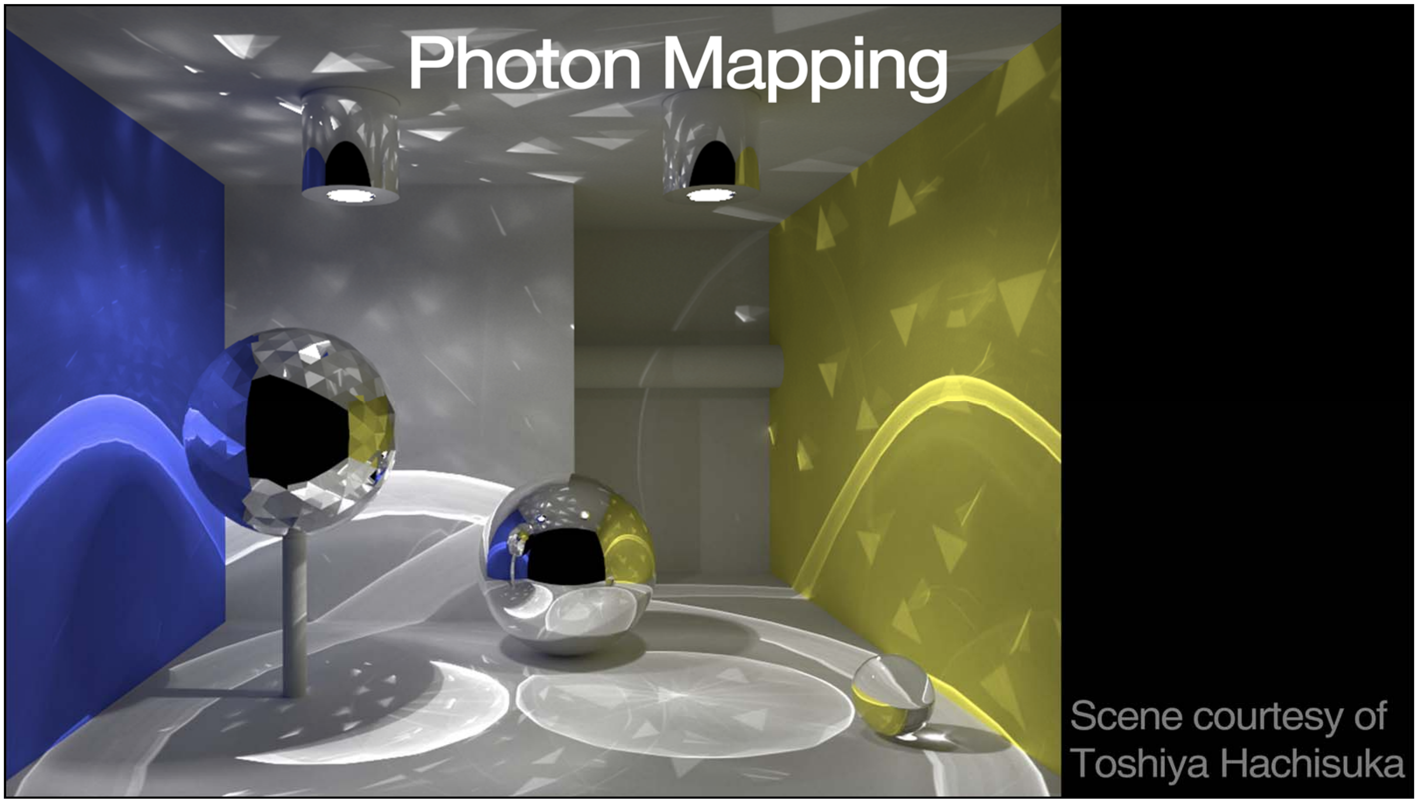


Unbiased methods have difficulties with rendering scenes which include so-called specular-diffuse-specular light paths. Such scenes are typical for realistic light settings, where the light source is not directly visible, but covered behind a refracting surface like a lens or reflected by a mirror.



Path Tracing

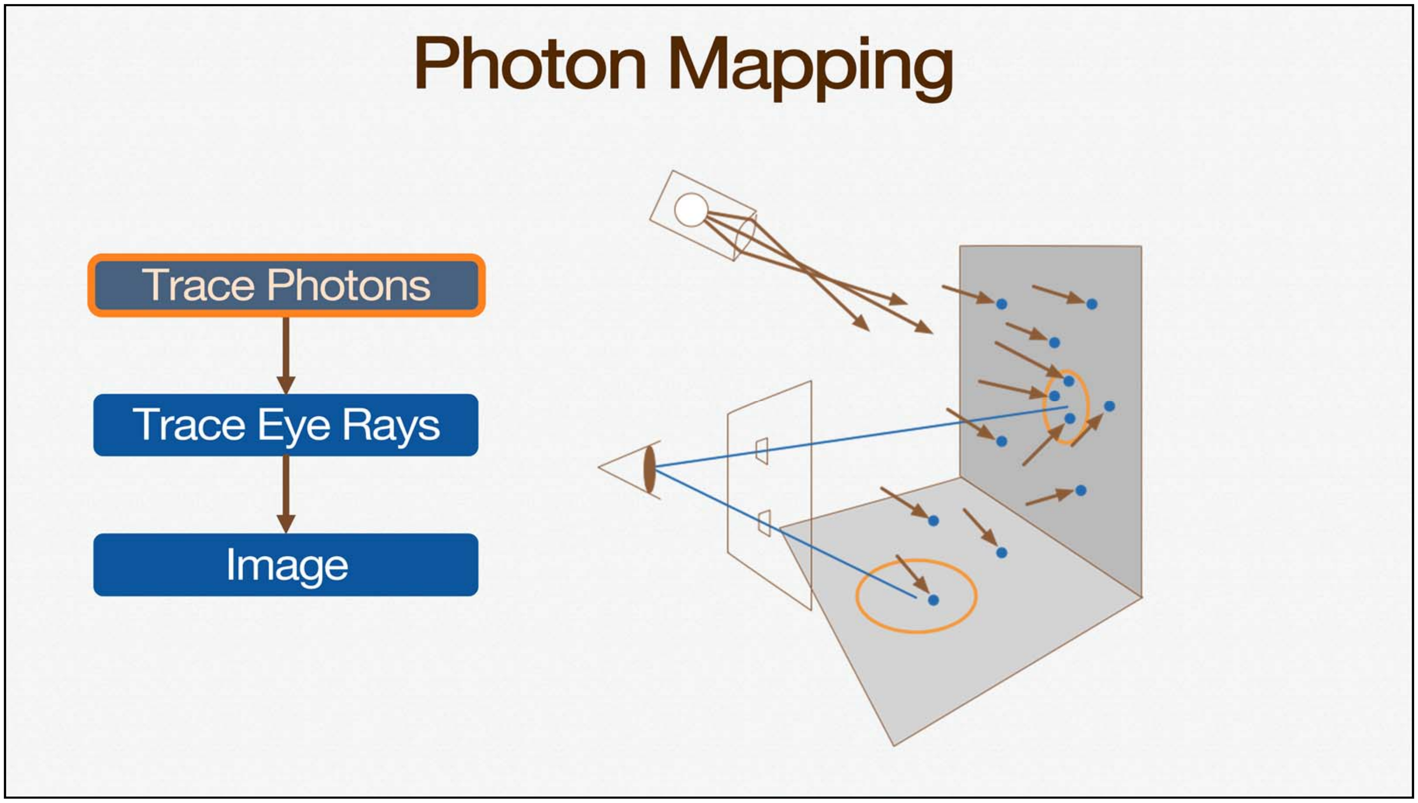
If we were to render such a scene using path tracing, we would see almost nothing. This is because the specular-diffuse-specular paths are sampled with probability close to 0.



Photon Mapping

Scene courtesy of Toshiya Hachisuka

This is the same scene rendered using photon mapping. Arguably, there is much less noise.



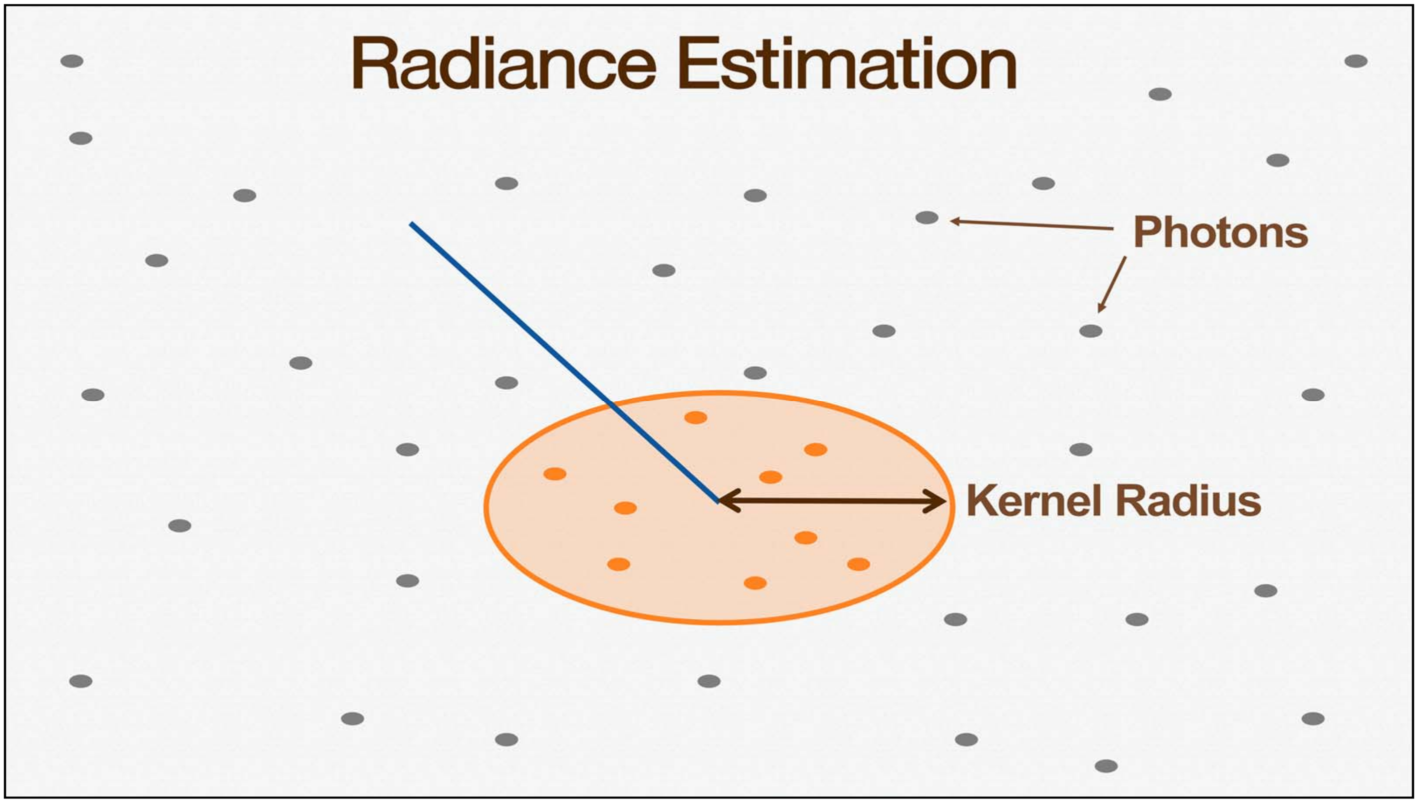
Photon Mapping

Trace Photons

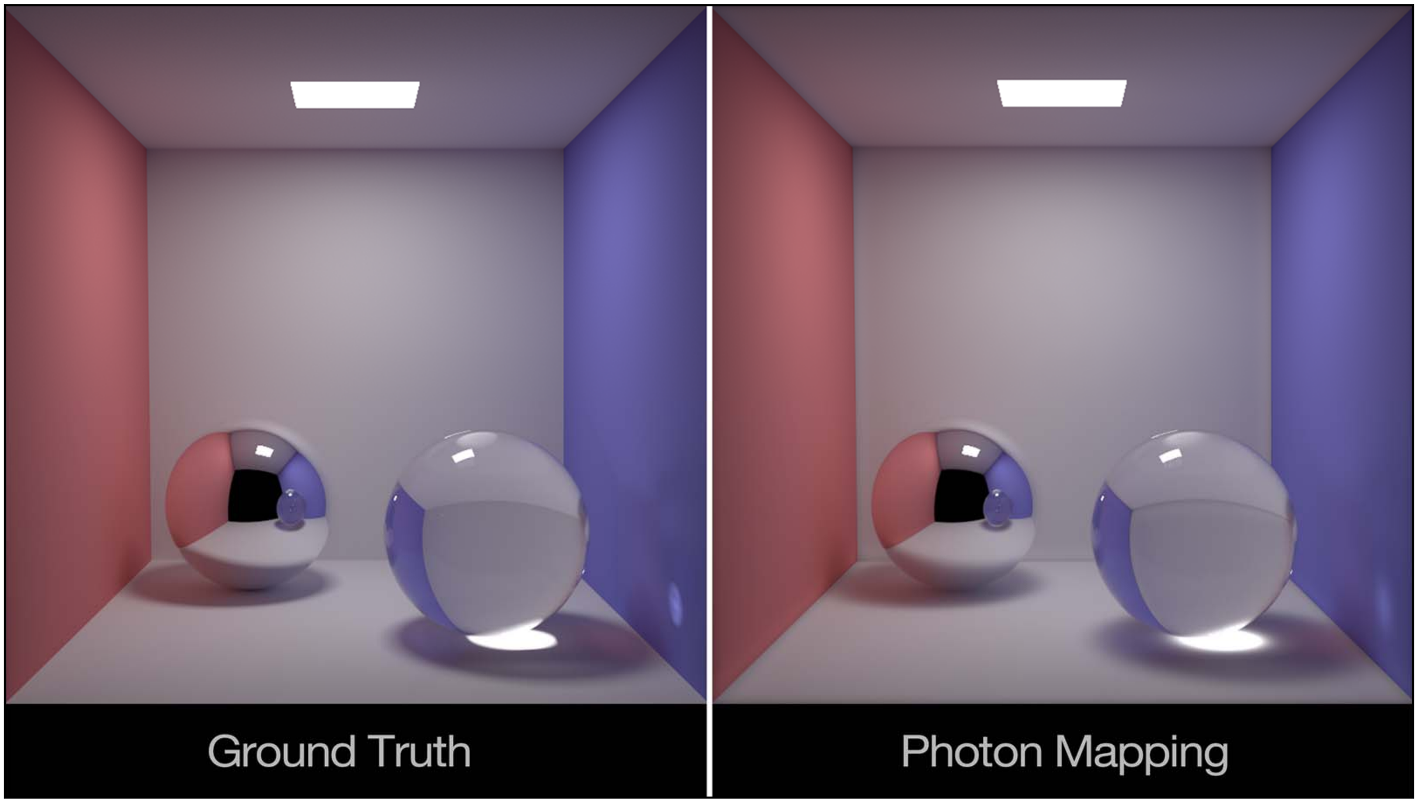
Trace Eye Rays

Image

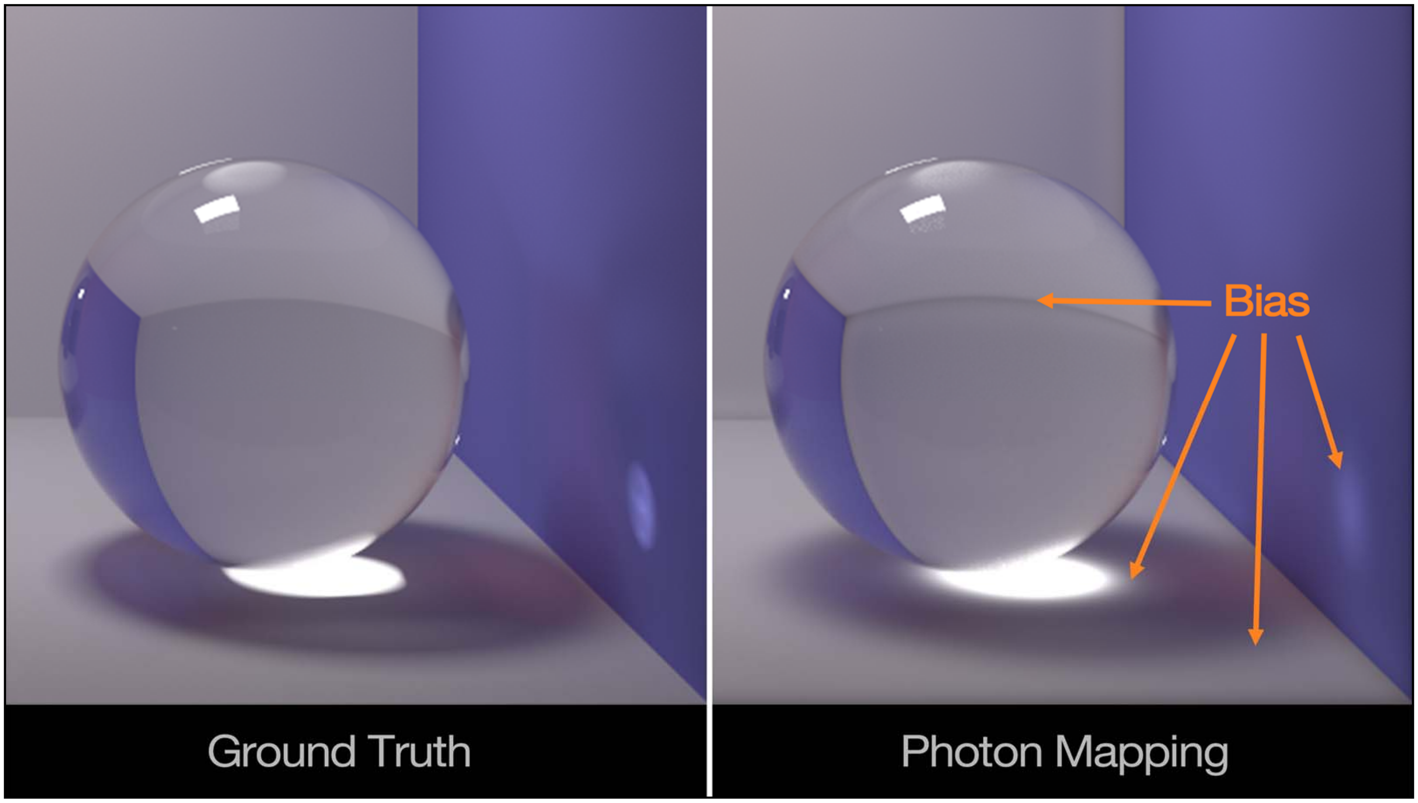
What makes photon mapping so good in rendering such scenes? Photon Mapping is a biased method. While unbiased methods sample entire paths from sensor to light, photon mapping samples partial paths. In a first step, it samples paths emitted from the light source and caches them as photons in a spatial data structure, the photon map. In the second step it samples paths from the eye and connects them to the previously cached light paths. It is this caching and reusing of light paths which makes photon mapping efficient.



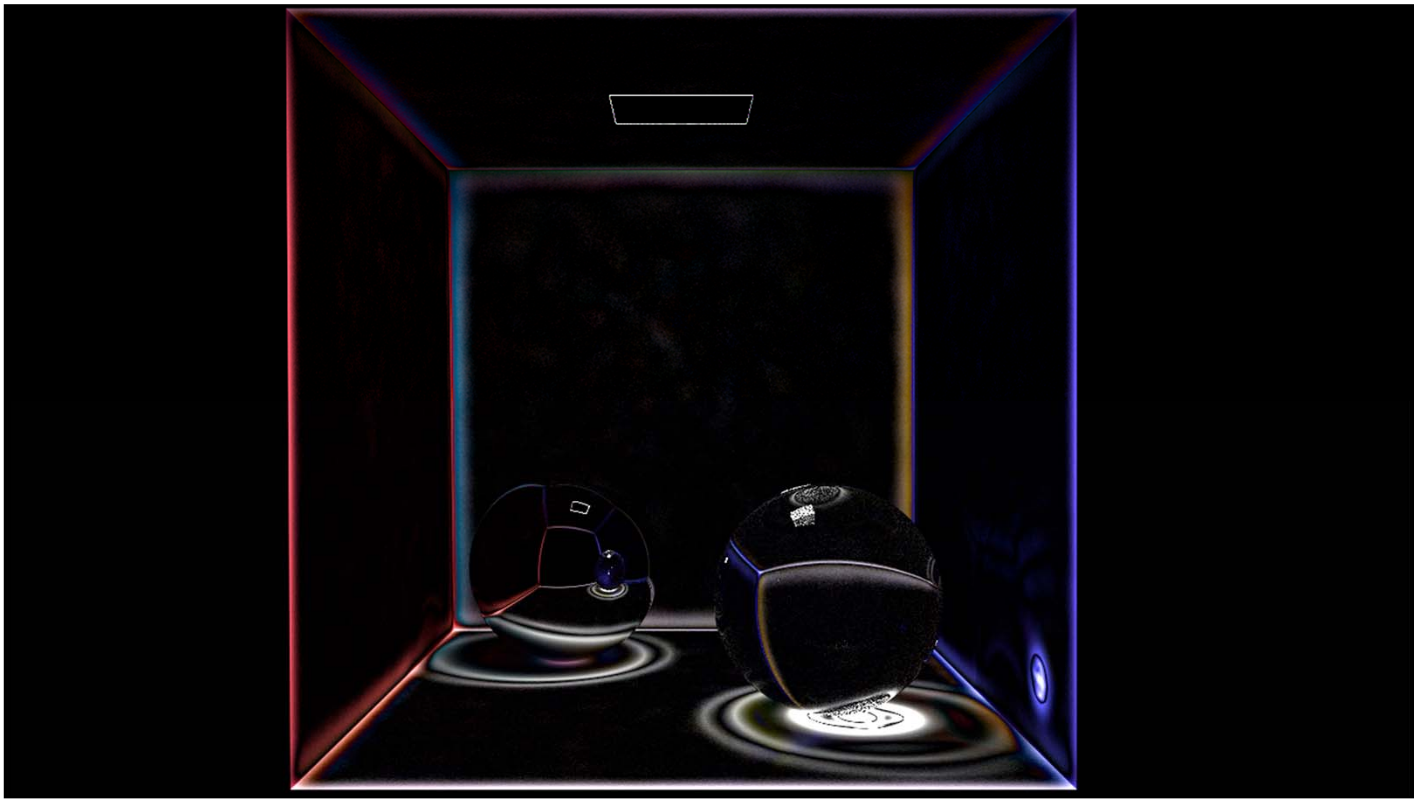
The partial paths are connected by so called radiance estimation. It involves the use of a kernel, which has a certain radius.



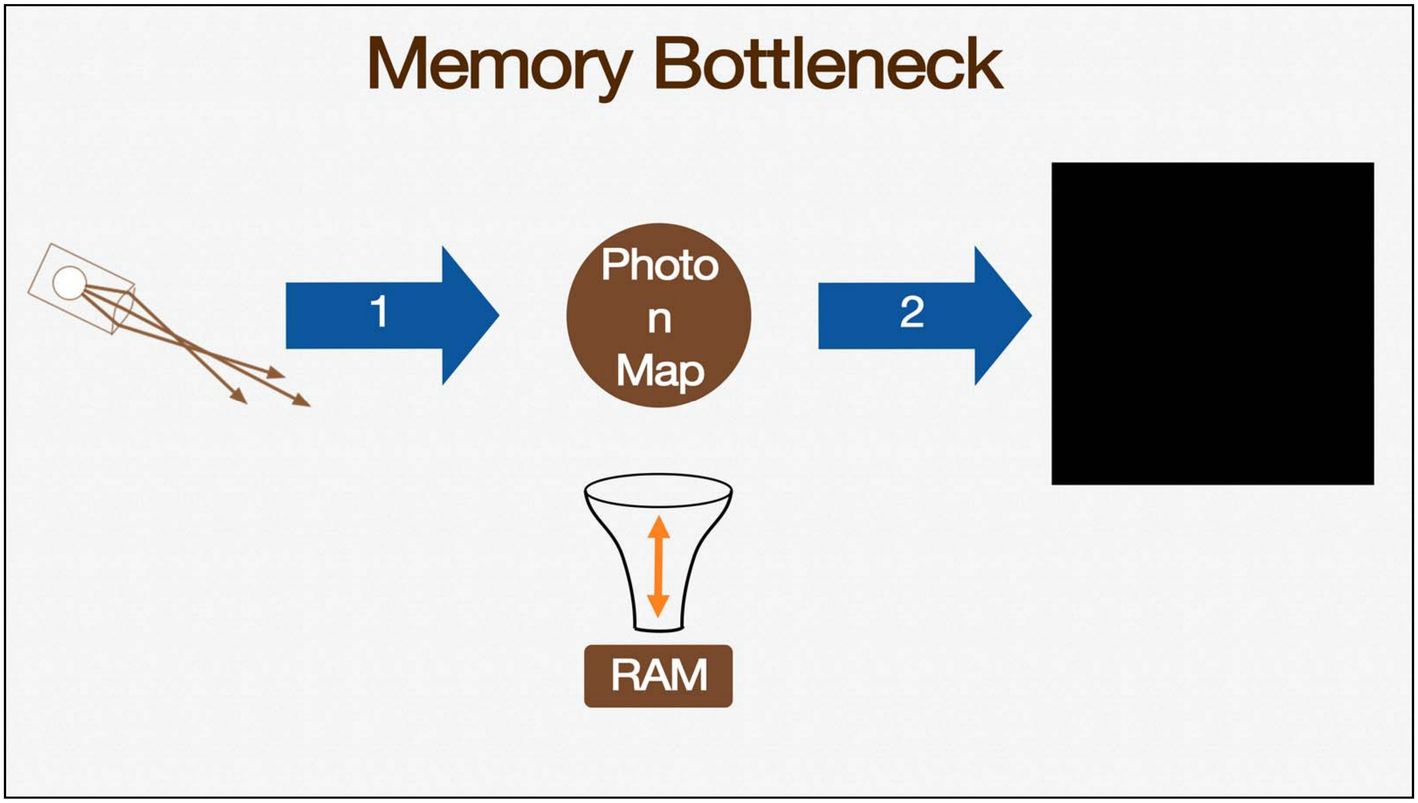
It is this radius which is the source of to the biggest critique of photon mapping, that it is biased.
Radiance estimation involves a convolution or interpolation, which often introduces blurriness, and therefore there is a consistent error, which is called bias.



It is this radius which is the source of to the biggest critique of photon mapping, that it is biased.
Radiance estimation involves a convolution or interpolation, which introduces blurriness, and therefore there is a consistent error, which is called bias.



Here is another image of the bias, shown as the difference between the two previous images.



Memory Bottleneck

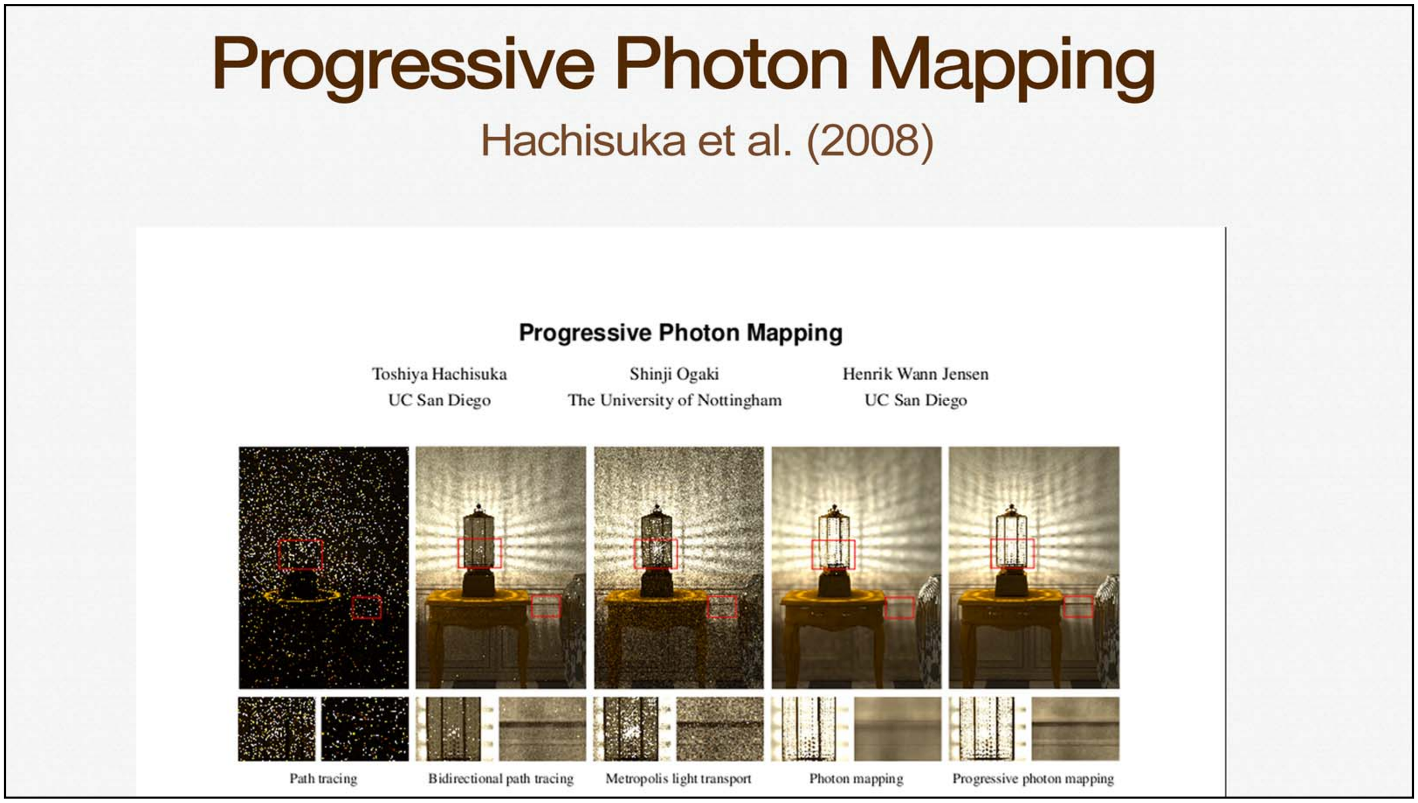
1

Photo
n
Map

2

RAM

Although photon mapping is biased, it is still a consistent method, which means that with an infinite number of samples, the bias vanishes and the resulting image is the correct. But here lies the practical problem of photon mapping: since the photons representing the light paths must be cached between the two stages, the quality of the image is limited by the available memory.



Progressive Photon Mapping

Hachisuka et al. (2008)

Progressive Photon Mapping

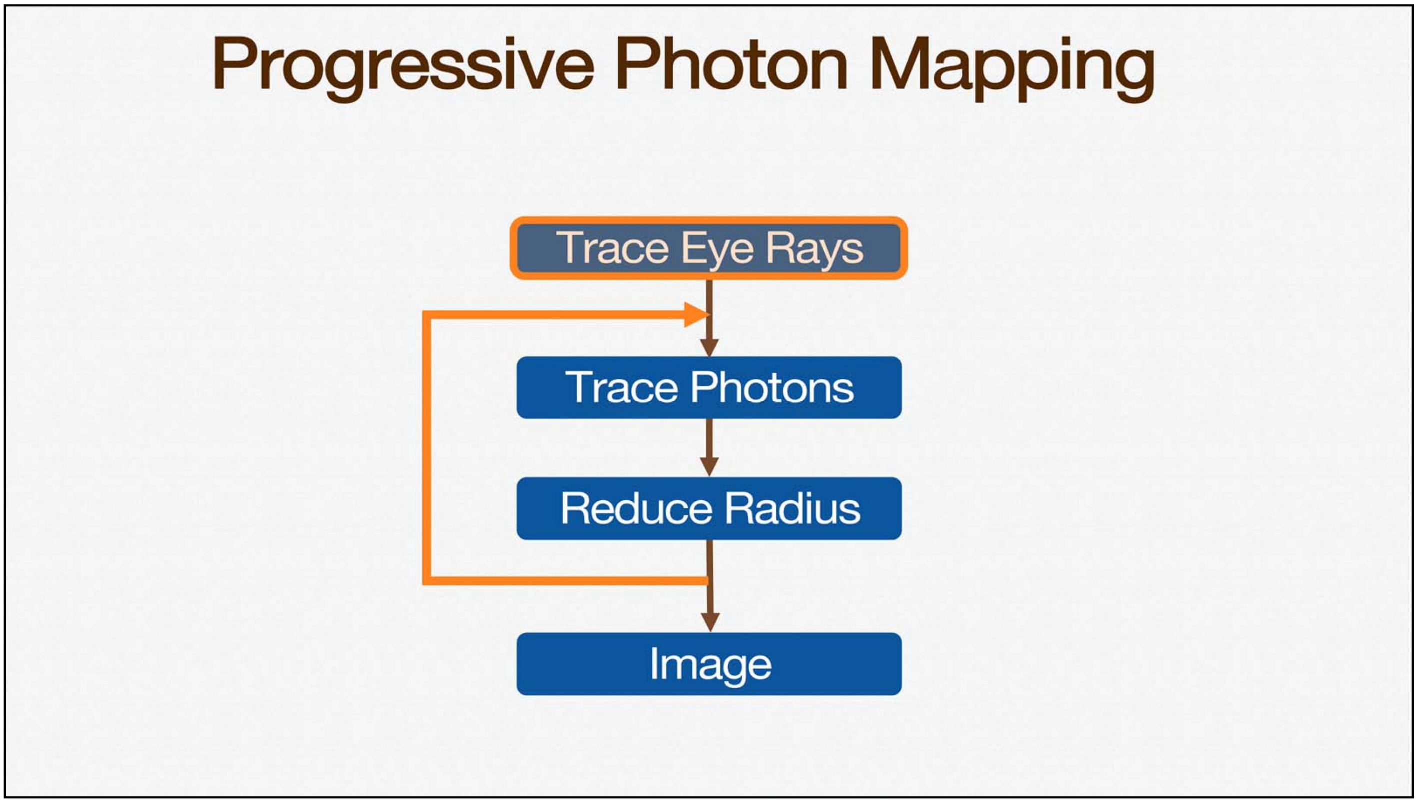
Toshiya Hachisuka
UC San Diego

Shinji Ogaki
The University of Nottingham

Henrik Wann Jensen
UC San Diego

Path tracing · Bidirectional path tracing · Metropolis light transport · Photon mapping · Progressive photon mapping

In 2008, Hachisuka et al. have solved this memory bottleneck with progressive photon mapping. The idea of progressive photon mapping is to update incrementally a sequence of photon mapping results using a limited number of photons at a time.



Progressive Photon Mapping

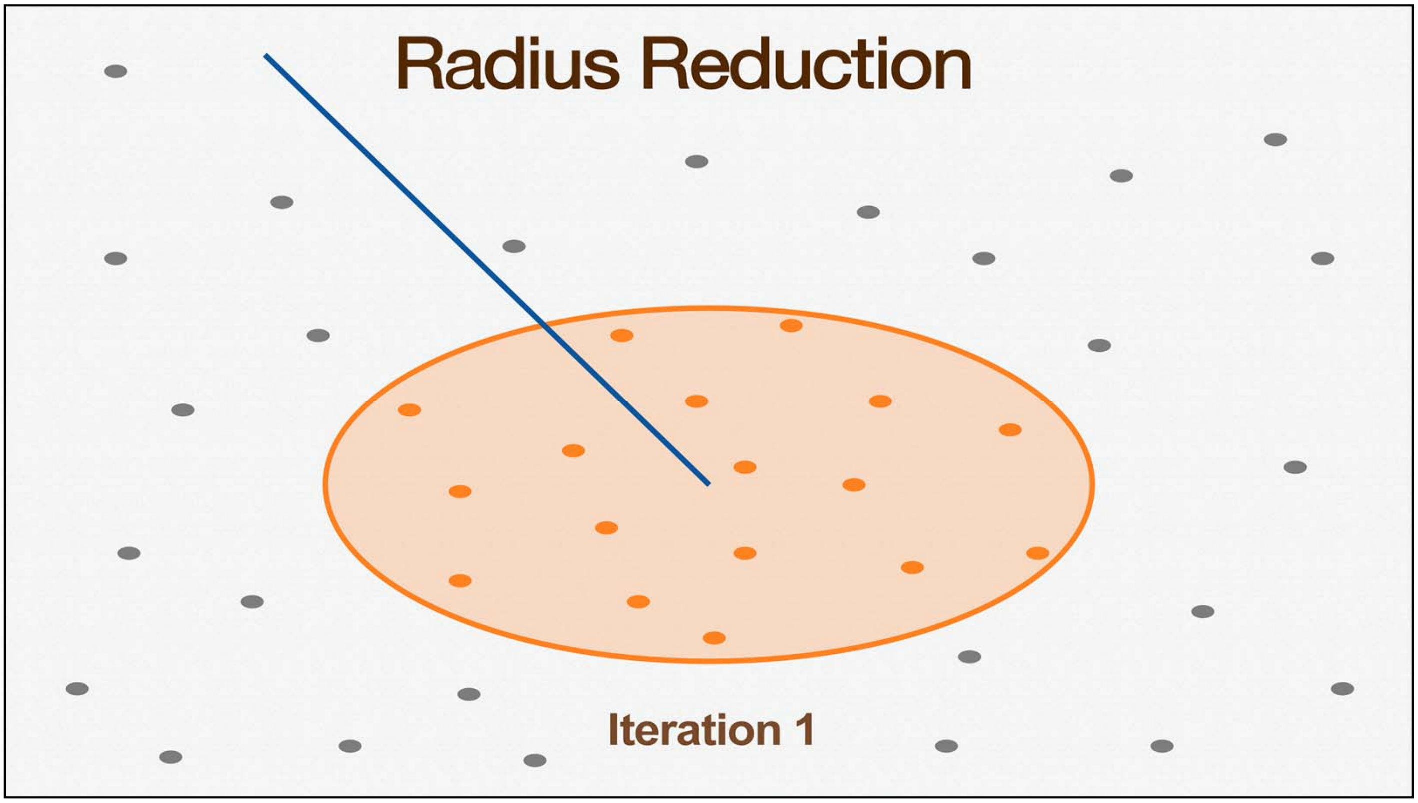
Trace Eye Rays

Trace Photons

Reduce Radius

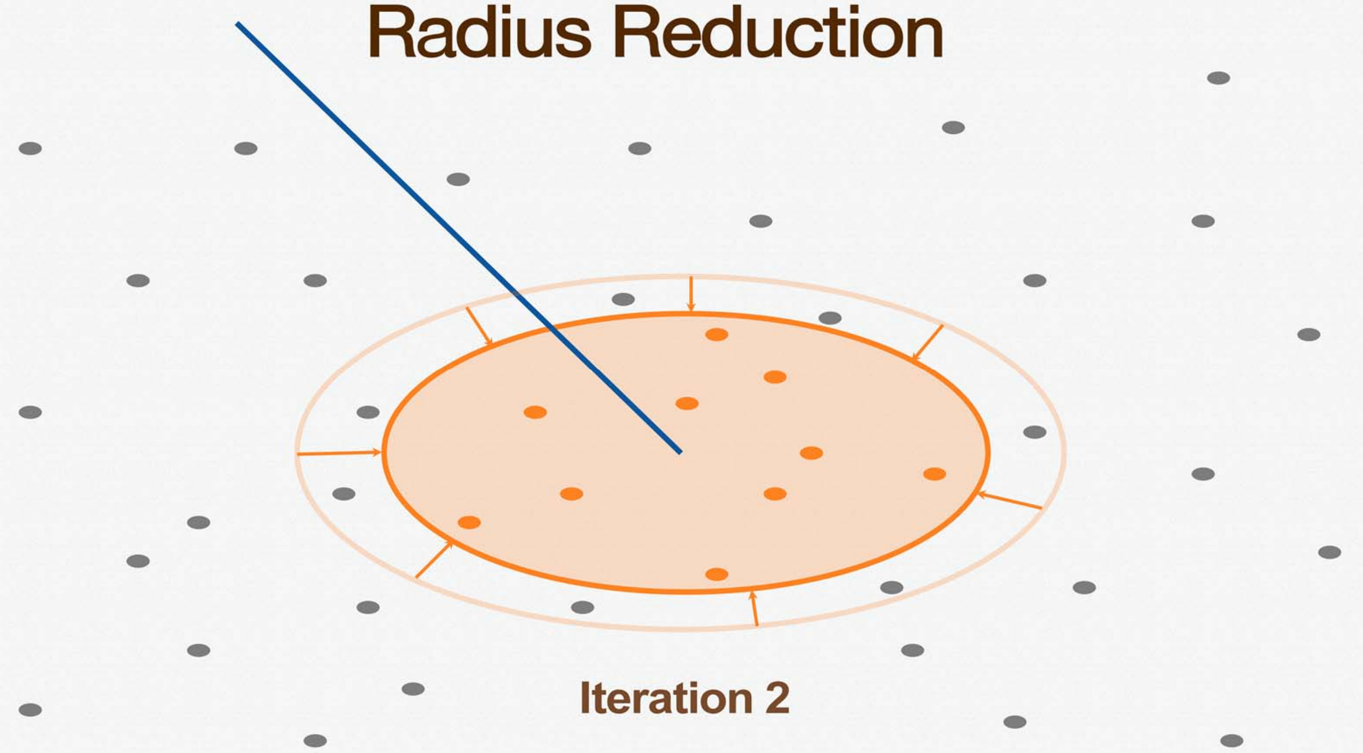
Image

Progressive photon mapping is an iterative method. Before the iteration, eye rays are traced and stored in locations. Then, for every iteration, new (independent) photons are traced. Finally, the most important step, is when the kernel radius is reduced; and then a new iteration begins.



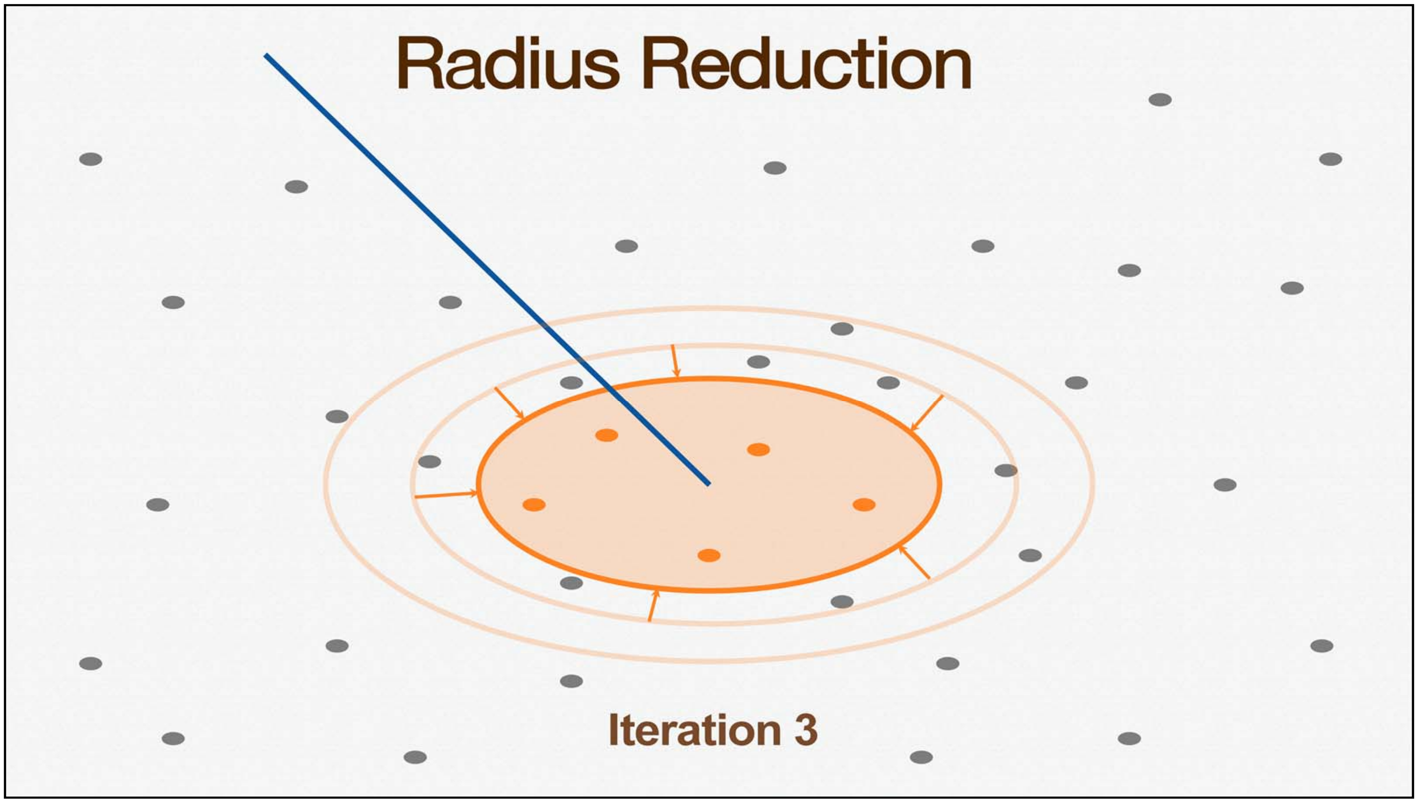
The key of progressive photon mapping is to reduce the kernel radius in every iteration, such that the bias vanishes.

Radius Reduction

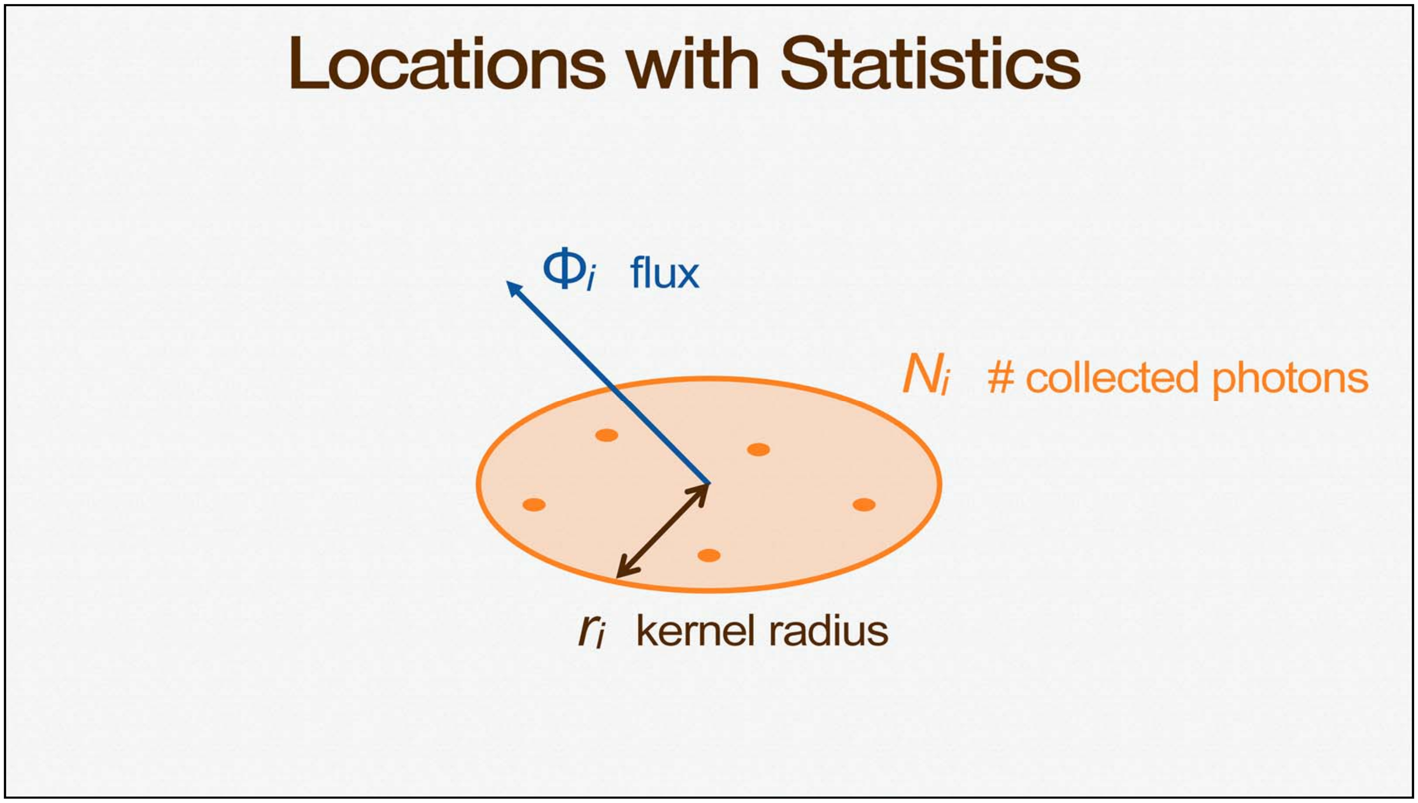


Iteration 2

Radius Reduction

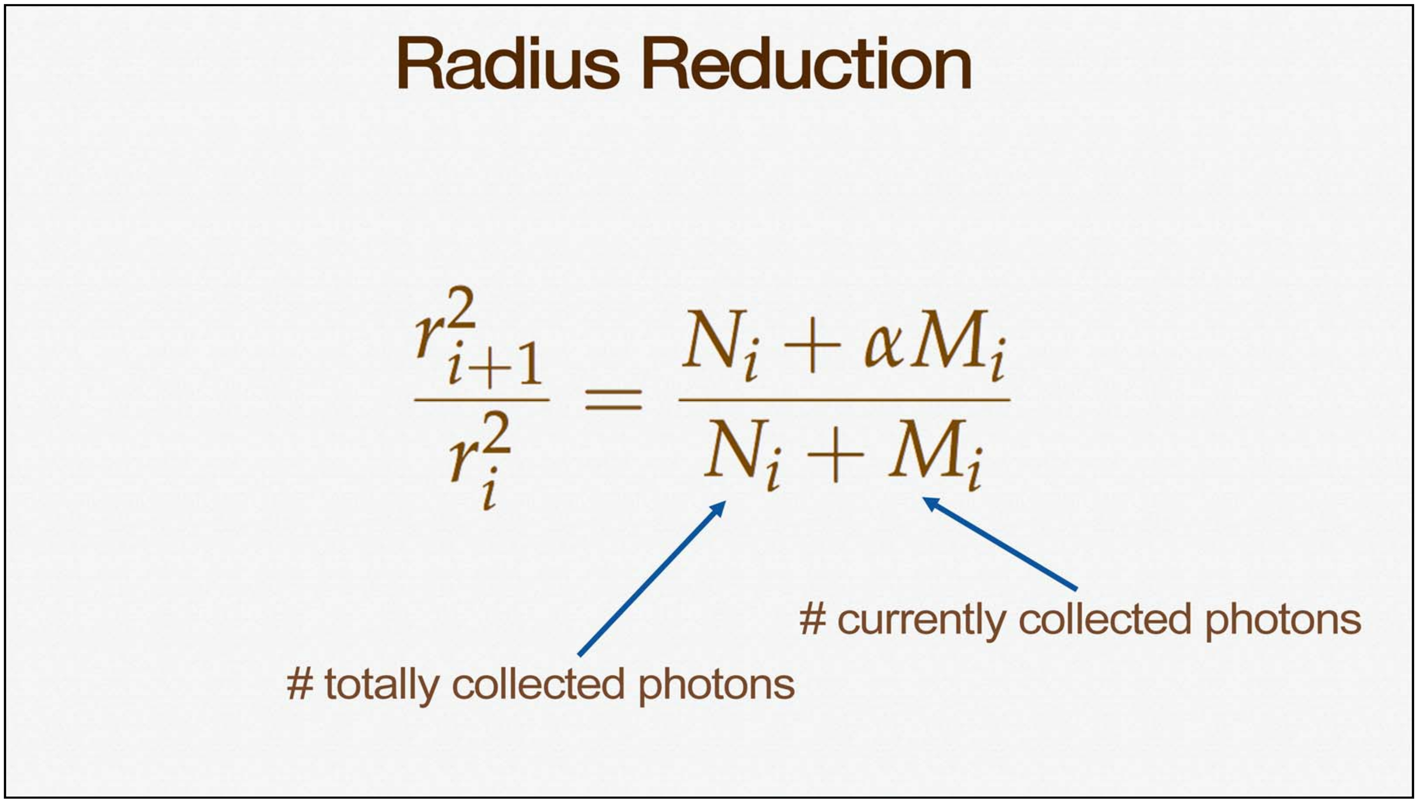


Iteration 3



Locations with Statistics

It does this by using statistics which are stored in every location of radiance estimation, namely the number of collected photons, the flux, and the kernel radius.



Radius Reduction

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

currently collected photons

totally collected photons

The exact radius reduction from iteration i to i + 1 is calculated using this update rule, which involves the use of these local statistics, such as the number of collected photons.

Stochastic PPM

Hachisuka & Jensen (2009)

Stochastic Progressive Photon Mapping

Toshiya Hachisuka Henrik Wann Jensen

UC San Diego

Figure 1: *Tools with a flashlight. The scene is illuminated by caustics from the flashlight, which cause SDS paths on the flashlight and highly glossy reflections of caustics on the bolts and plier. The flashlight and the plier are out of focus. Using the same rendering time, our method (right) robustly renders the combination of the complex illumination setting and the distributed ray tracing effects where progressive photon mapping is inefficient (left).*

Abstract

This paper presents a simple extension of progressive photon mapping for simulating global illumination with effects such as depth-of-field, motion blur, and glossy reflections. Progressive photon mapping is a robust global illumination algorithm that can handle complex illumination settings including specular-diffuse-specular paths. The algorithm can compute the correct radiance value at a point in the limit. However, progressive photon mapping is not effective at rendering distributed ray tracing effects, such as depth-of-field, that requires multiple pixel samples in order to compute the correct average radiance value over a region. In this paper, we introduce a new formulation of progressive photon mapping, called stochastic progressive photon mapping, which makes it possible to compute the correct average radiance value for a region. The key idea is to use shared photon statistics within the region rather than isolated photon statistics at a point. The algorithm is easy to implement, and our results demonstrate how it efficiently handles scenes with distributed ray tracing effects, while maintaining the robustness of progressive photon mapping in scenes with complex lighting.

1 Introduction

Efficiently computing global illumination is an active areas of research in computer graphics. The types of lighting can vary significantly in different scenes, and it is important to develop algorithms that can handle this variation robustly.

Global illumination algorithms solve the rendering equation introduced by Kajiya [1986]. Unbiased Monte Carlo methods have been a popular approach for computing global illumination without any approximations in the past few decades [Dutré et al. 2006]. However, unbiased methods are not robust under all illumination settings. There are certain light paths that are problematic. For example, path tracing [Kajiya 1986] works well for a scene with diffuse materials, however, it cannot efficiently handle caustics from a small light source.

Hachisuka et al. [2008] observed out that specular-diffuse-specular paths (SDS paths in the light path notation) are particularly problematic for the existing unbiased methods. An example of an SDS path is a light path due to the combination of specular materials and a light bulb. Photon mapping [Jensen 1996] is a biased method which is robust in the presence of SDS paths. However, the results suffer from bias, which appears as low frequency noise in the rendered images. Moreover, computing the correct solutions requires storing an infinite number of photons in the limit. Progressive photon mapping [Hachisuka et al. 2008] solves this issue by using progressive refinement, and makes it possible to compute a correct solutions without storing any photons. Moreover, the method retains the robustness of photon mapping.

Although each radiance estimate in progressive photon mapping converges to the correct radiance, the algorithm is restricted to computing the correct radiance value at a point. This property limits the applications of progressive photon mapping because we often need to compute the correct average radiance value over a region. For example, anti-aliasing in ray tracing requires the average radiance value for a pixel footprint. Depth-of-field is another example where each pixel value is the average radiance value for the part of a scene that is visible through the lens. In general, all the effects that can be achieved by distributed ray tracing [Cook et al. 1984] require computing the average radiance over some domain.

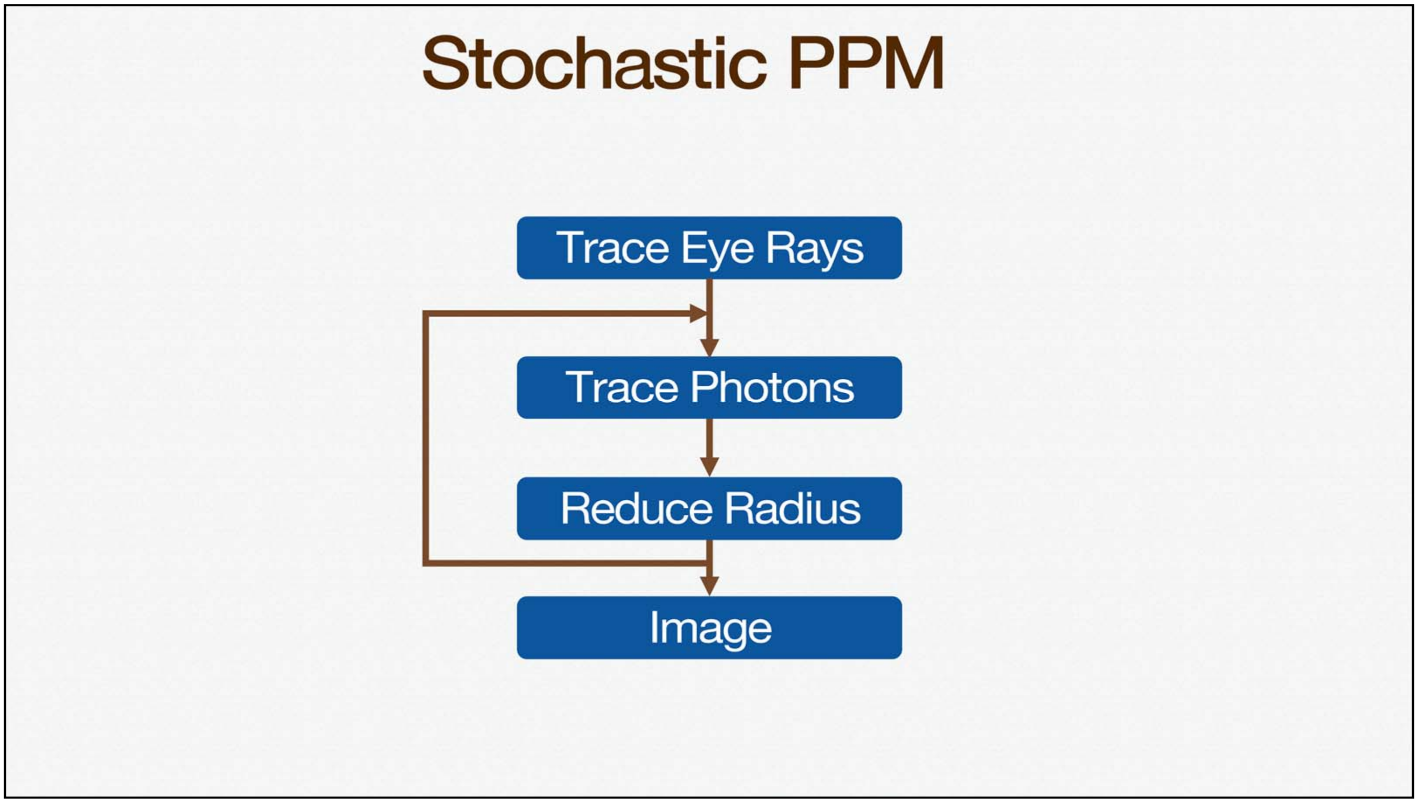
In this paper, we present a new formulation of progressive photon mapping that enables computing the correct average radiance value over a region. Our formulation requires a simple algorithmic modification, which consists of adding a distributed ray tracing pass after each photon pass in progressive photon mapping. The main contribution is this new formulation that allows simple, yet effective improvement of the robustness of the progressive photon mapping. We show that our modification allows us to render scenes with distributed ray tracing effects in combination with complex illumination scenarios.

★ Glossy reflections

★ Depth of field

★ Motion blur

In 2009, in a follow up work, Hachisuka and Jensen have generalized the method to stochastic progressive photon mapping. It includes additional effects like glossy reflections, depth of field, and motion blur.



Stochastic PPM

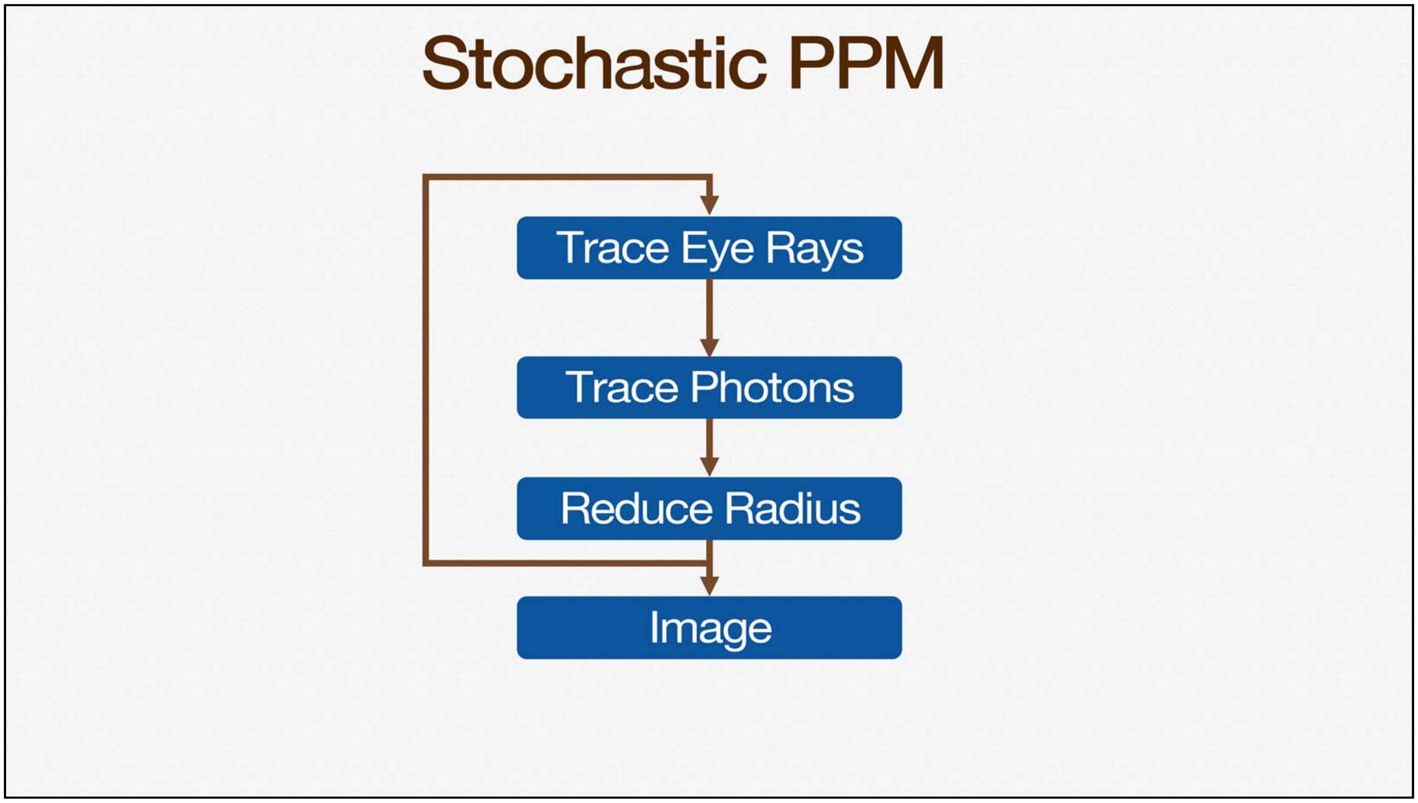
Trace Eye Rays

Trace Photons

Reduce Radius

Image

The main change of stochastic progressive photon mapping is that instead of tracing eye rays once, the rays are traced in every iteration.



Stochastic PPM

Trace Eye Rays

Trace Photons

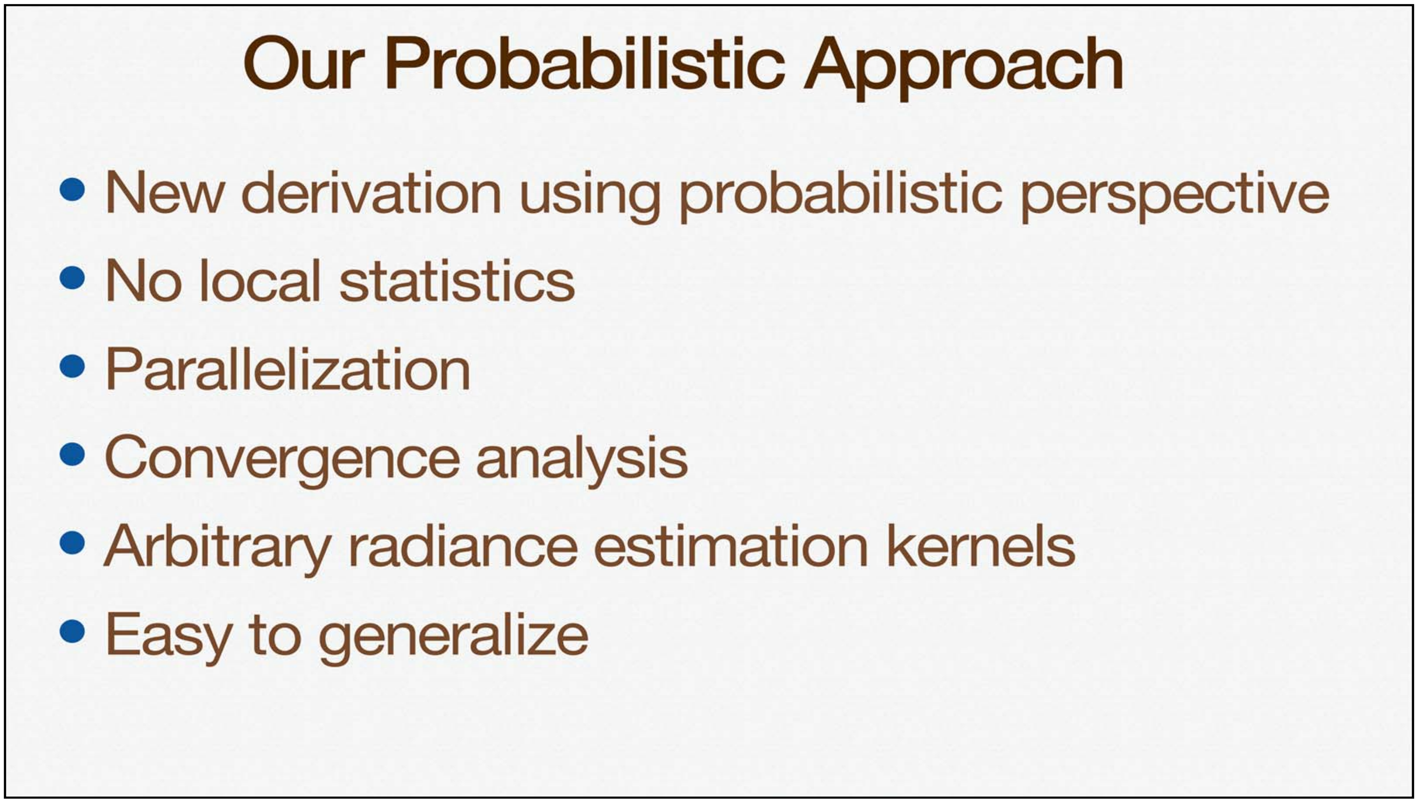
Reduce Radius

Image

The main change to progressive photon mapping is that instead of tracing eye rays once, they are traced in every iteration.

Great, but ...

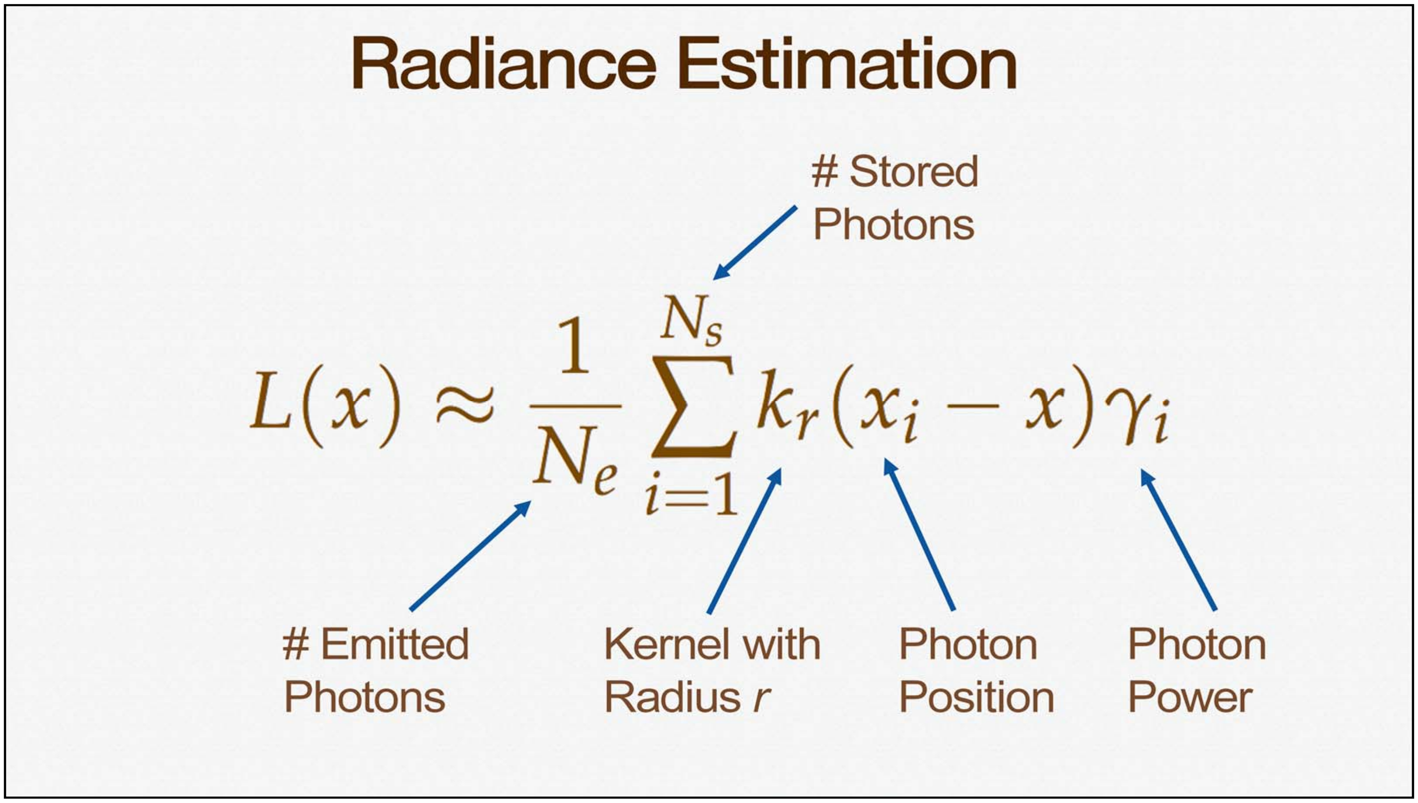
While progressive photon mapping is great, we wanted to make a step back and look at it from a different perspective.



Our Probabilistic Approach

- New derivation using probabilistic perspective
- No local statistics
- Parallelization
- Convergence analysis
- Arbitrary radiance estimation kernels
- Easy to generalize

We found a new derivation of progressive photon mapping which does not require local statistics and is trivial to parallelize. We also provide convergence analysis. Furthermore, our reformulation of progressive photon mapping generalizes to arbitrary radiance estimation kernels. And finally, it is easy to generalize to other radiance estimates like volumetric photon mapping and the recent work of beam radiance estimates.



Radiance Estimation

Stored Photons

$$L(x) \approx \frac{1}{N_e} \sum_{i=1}^{N_s} k_r(x_i - x)\gamma_i$$

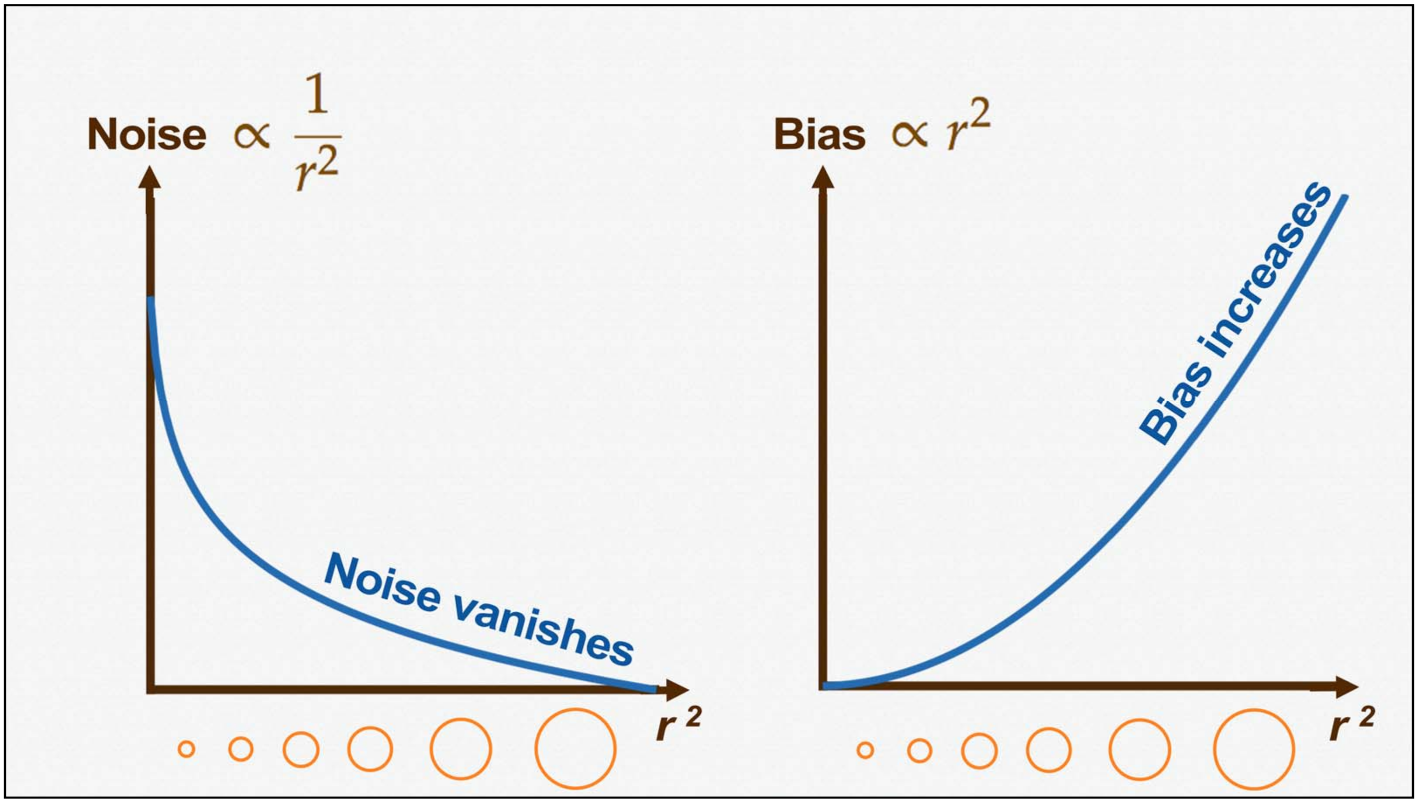
Emitted Photons

Kernel with Radius *r*

Photon Position

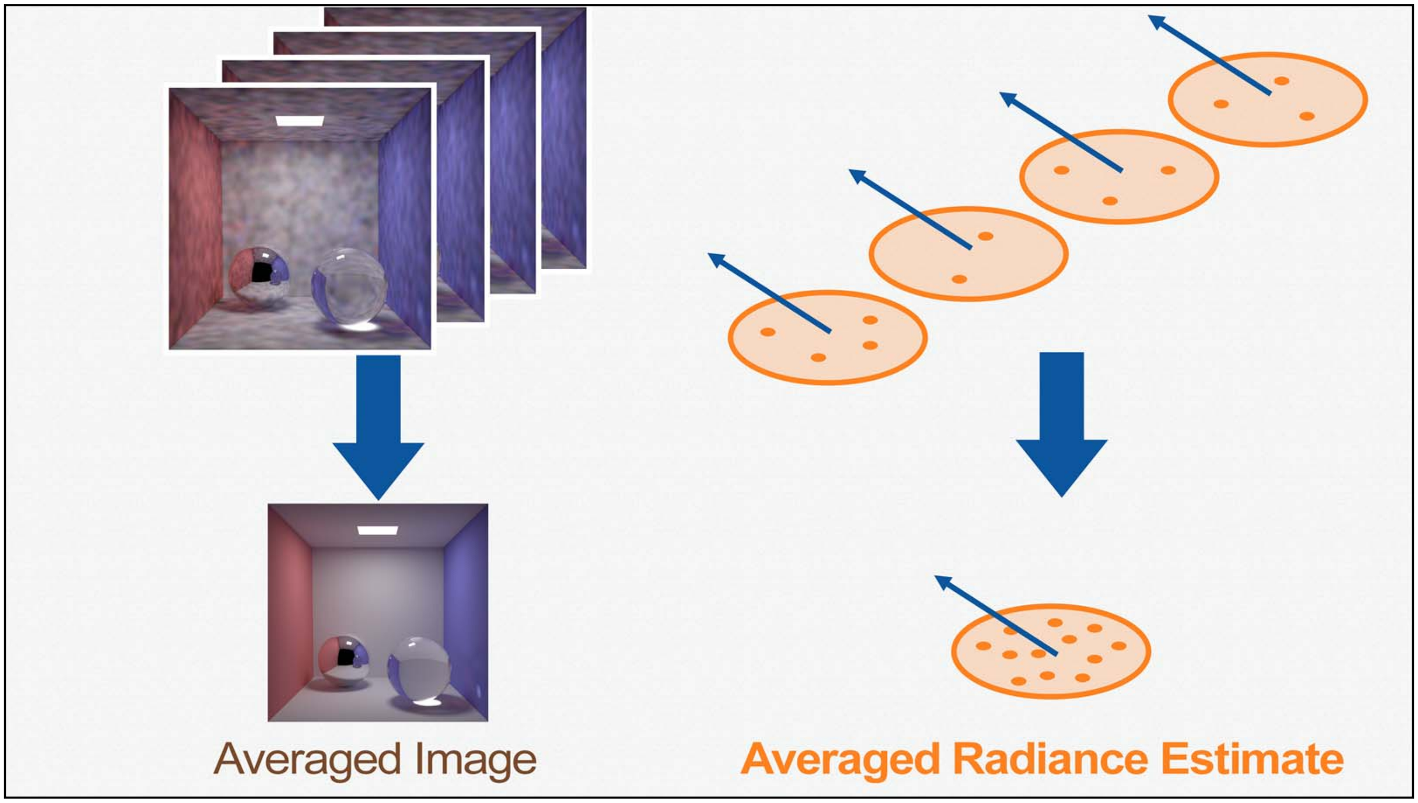
Photon Power

Let us start with a probabilistic analysis of the radiance estimation. A radiance estimate is a Monte Carlo integral. N_e is the number of emitted photons. N_s is the number of stored photons. k_r is a kernel with radius r. x_i is the position of the photon and gamma_i is the power of the photon, already pre-multiplied with the BRDF, which amounts to the reflected radiance. Since a Monte Carlo integral is a stochastic method, we can look at the statistics of radiance estimation.

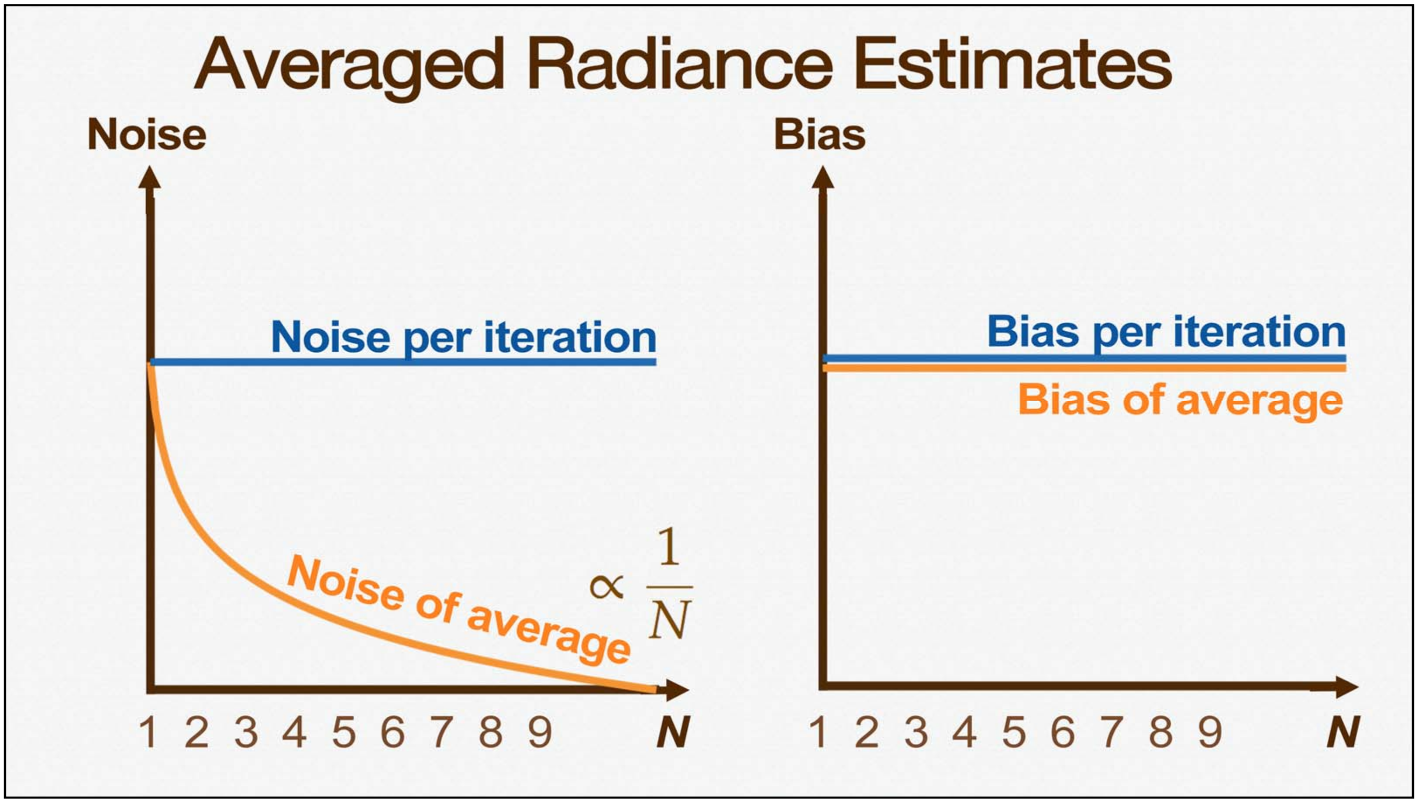


Namely, the noise and the bias. It turns out that the noise of the radiance estimate is inverse proportional to the squared radius. Intuitively, it is clear that increasing the radius has the effect of averaging over more photons, and will therefore reduce the noise.

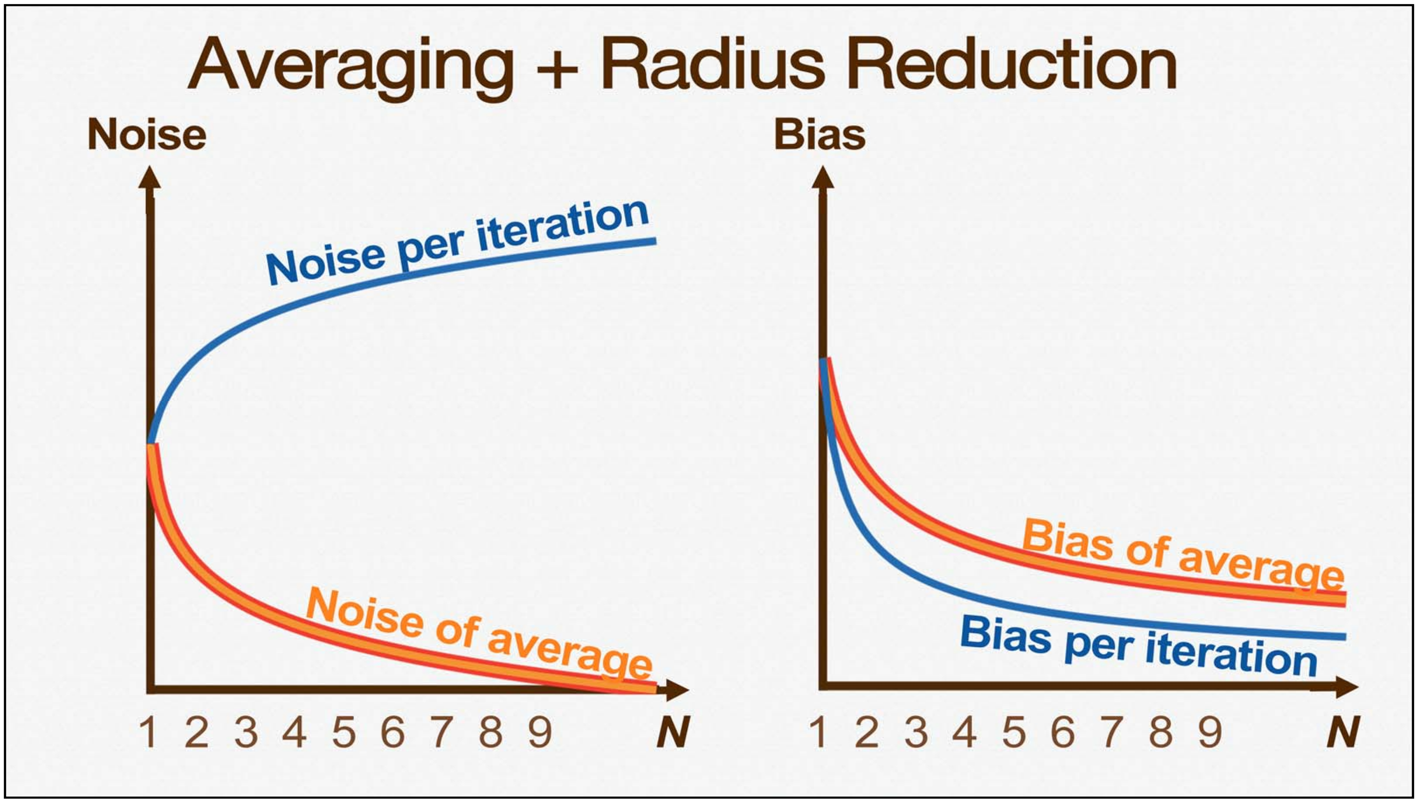
The bias, on the other hand, is proportional to the squared radius. In this case, increasing the radius will include more features like caustics, and therefore will increase the bias. Note that the bias only vanishes if the radius is 0. For any fixed radius, there is this classic trade-off where we can only achieve smoothness or unbiasedness but not both at the same time.



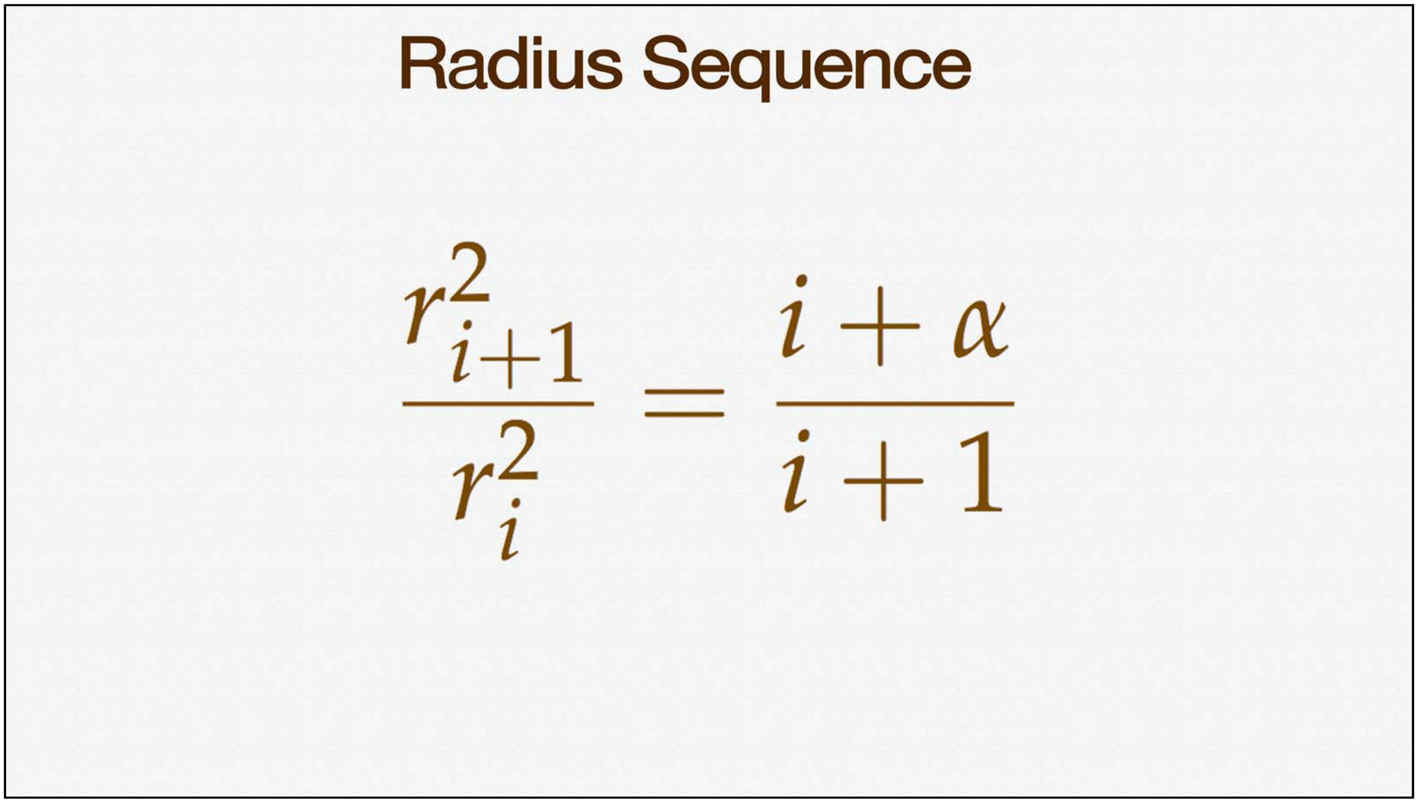
Following the idea of the original progressive photon mapping, we would like to split up the rendering into multiple iterations in order to remove the memory bottleneck. The easiest way to combine the iterations is to average the images. We show in the paper that the averaged image converges if the averaged radiance estimates converge. Let's see what happens if we average the radiance estimate over many iterations.



While the noise per iteration remains constant, the noise of the average will vanish with 1/N and the averaged image becomes smoother. This is great -- but how about the bias? The bias per iteration remains constant, and unfortunately, the bias of the averaged radiance estimate remains constant as well. This is not surprising, because we are simply averaging images, and this will reduce the noise, but not the bias.



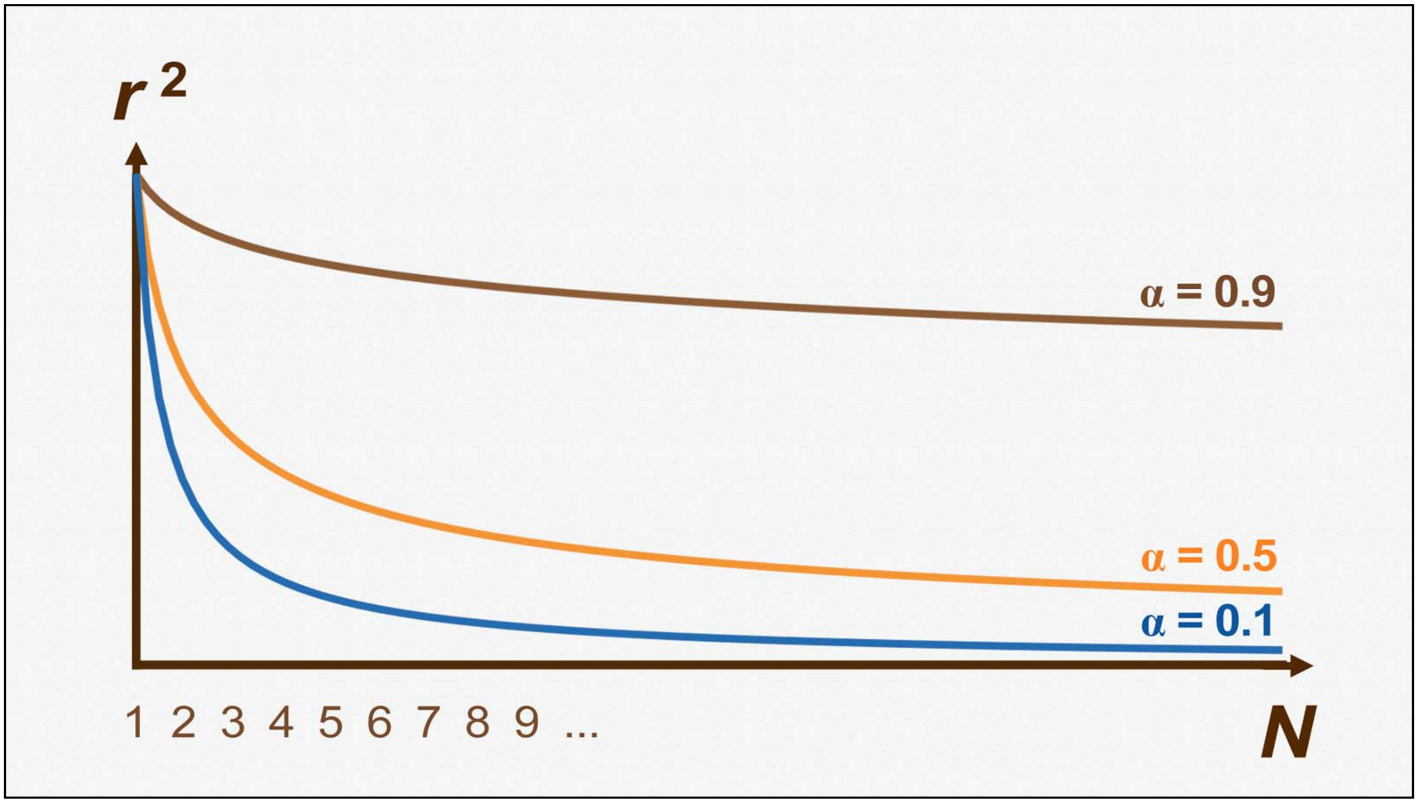
The only way to reduce the bias of the average is by reducing the kernel radius. If the radius is continuously reduced, not only the bias per iteration, but also the the bias of the average vanishes. But, if we reduce the radius, as we have seen before, the noise per iteration will increase. The trick of progressive photon mapping is to let the radius decrease slow enough such that the noise of the *averaged* radiance estimate still vanishes in the limit. To summarize: we look for a radius sequence which reduces slow enough for both the noise and bias of the average to vanish in the limit.



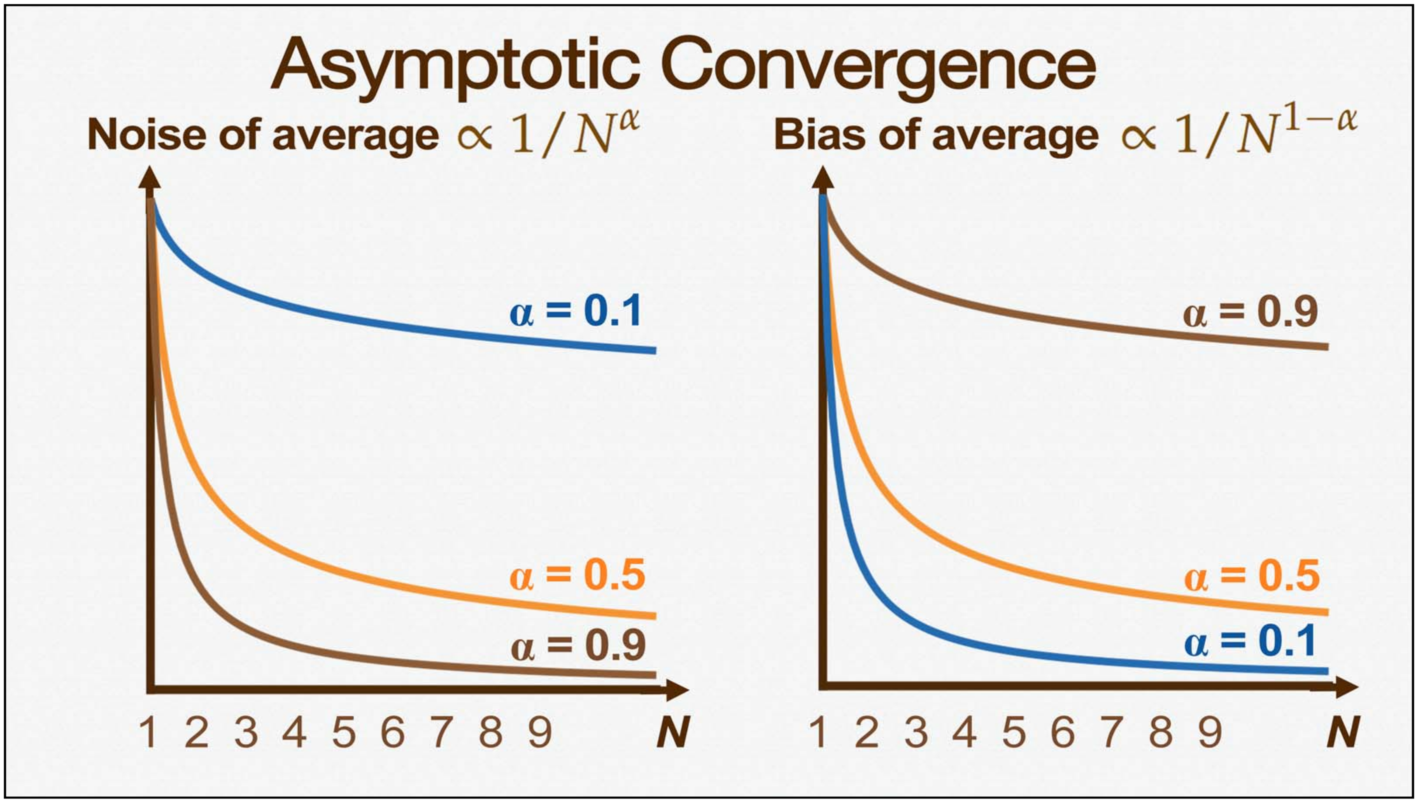
Radius Sequence

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

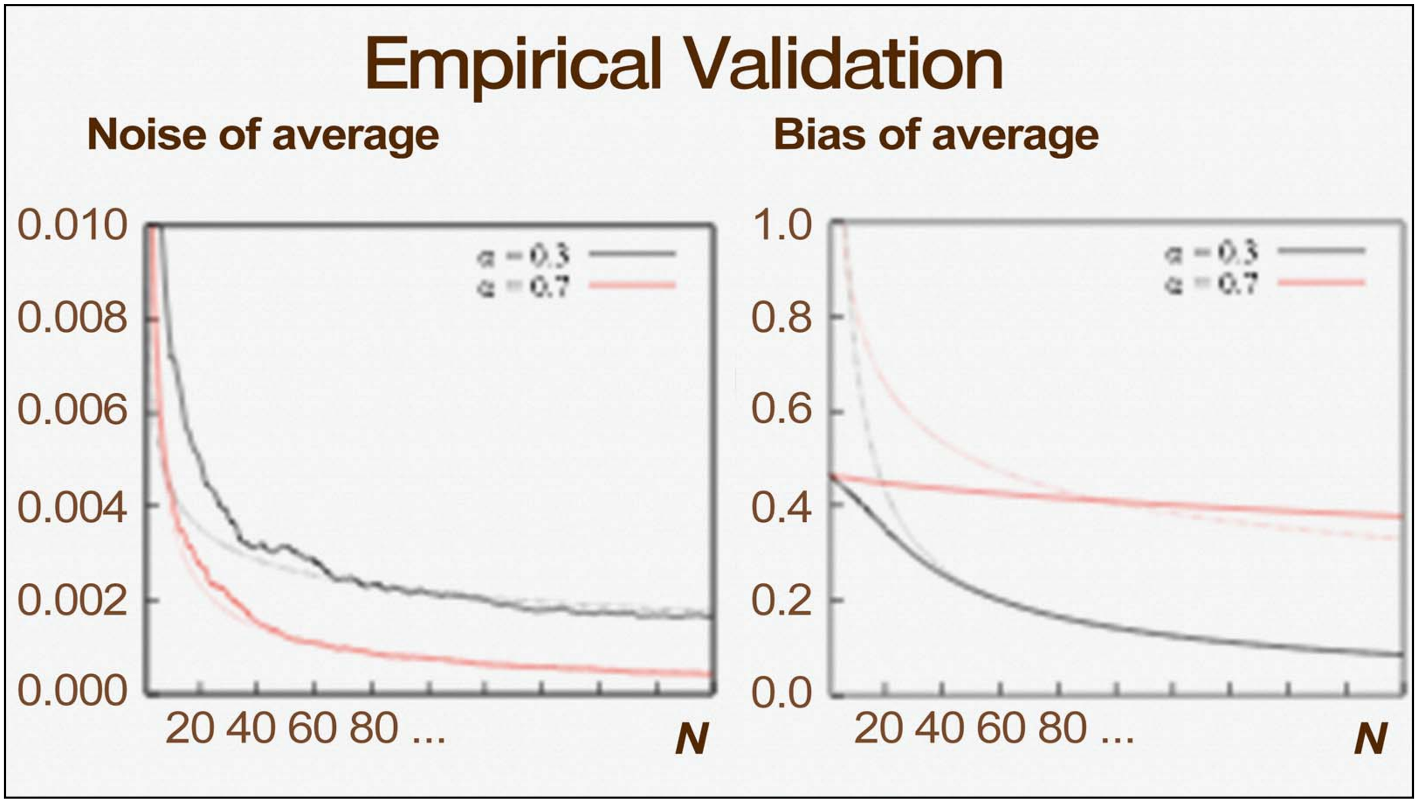
The radius sequence which allows us to walk this fine line where both noise and bias of averaged radiance estimates vanish in the limit is given by this formula. A detailed proof can be found in our paper. Our proposed radius sequence is defined recursively. i is the iteration number, and alpha is a parameter which must be between 0 and 1.



The parameter alpha controls how fast the radius is reduced with the number of iterations.



Using our proposed radius sequence, we found the following asymptotic convergence. The noise of the averaged radiance estimate vanishes proportionally to 1/N^alpha, where N is the number of iterations. The bias of the averaged radiance estimate vanishes proportionally to 1/N^(1-alpha). We can see here how alpha controls the convergence speed of noise and bias. A small alpha value like the blue curve reduces the noise slowly, but the bias will vanish quickly. A large alpha on the other hand, the brown curve, reduces the noise quickly, but the bias will go down slowly.

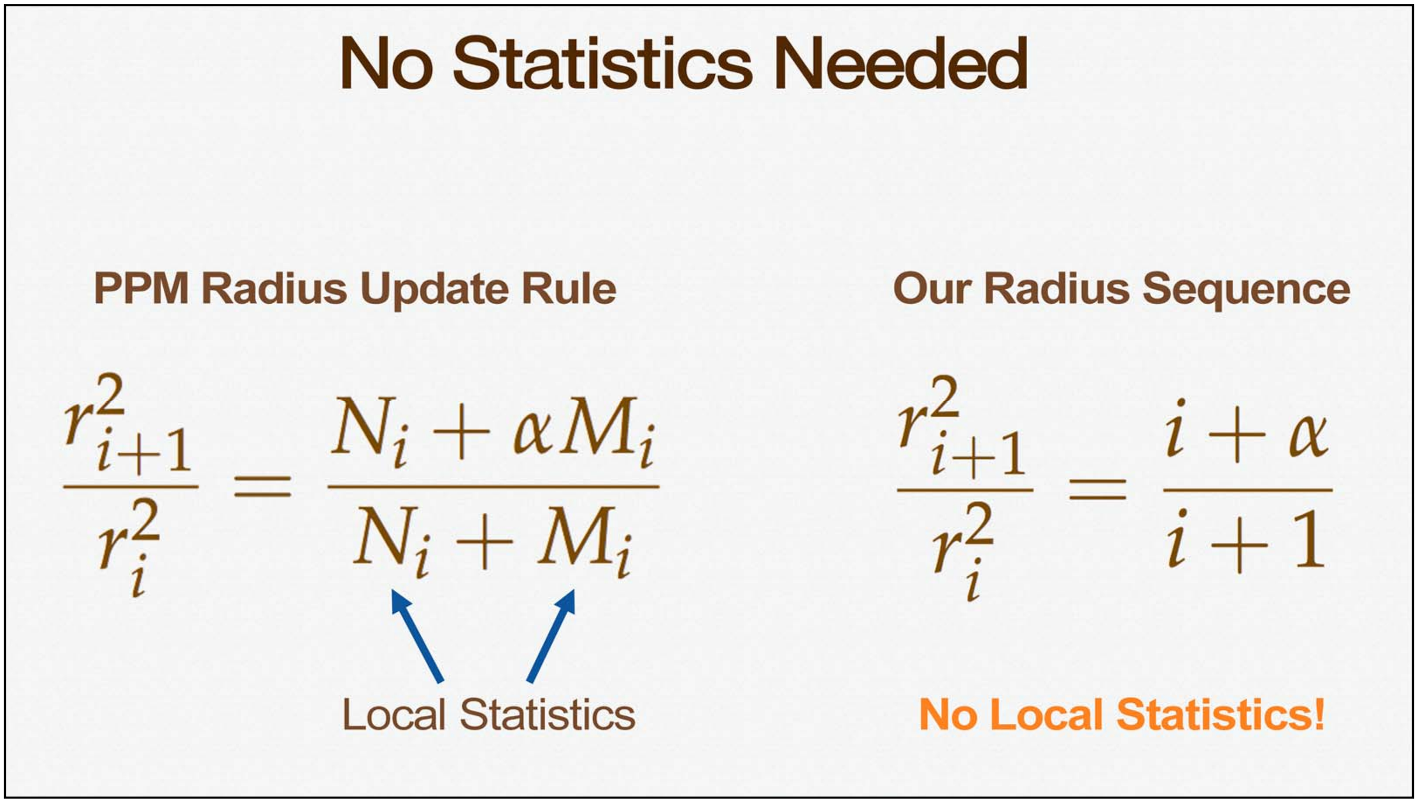


Empirical Validation

Noise of average

Bias of average

We have also empirically verified this asymptotic convergence. We see the convergence for a sequence of radiance estimates taken in the previous scene. Here the solid lines are the measured noise and bias for two different alpha values. The dotted lines are the asymptotic curves.



No Statistics Needed

PPM Radius Update Rule

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

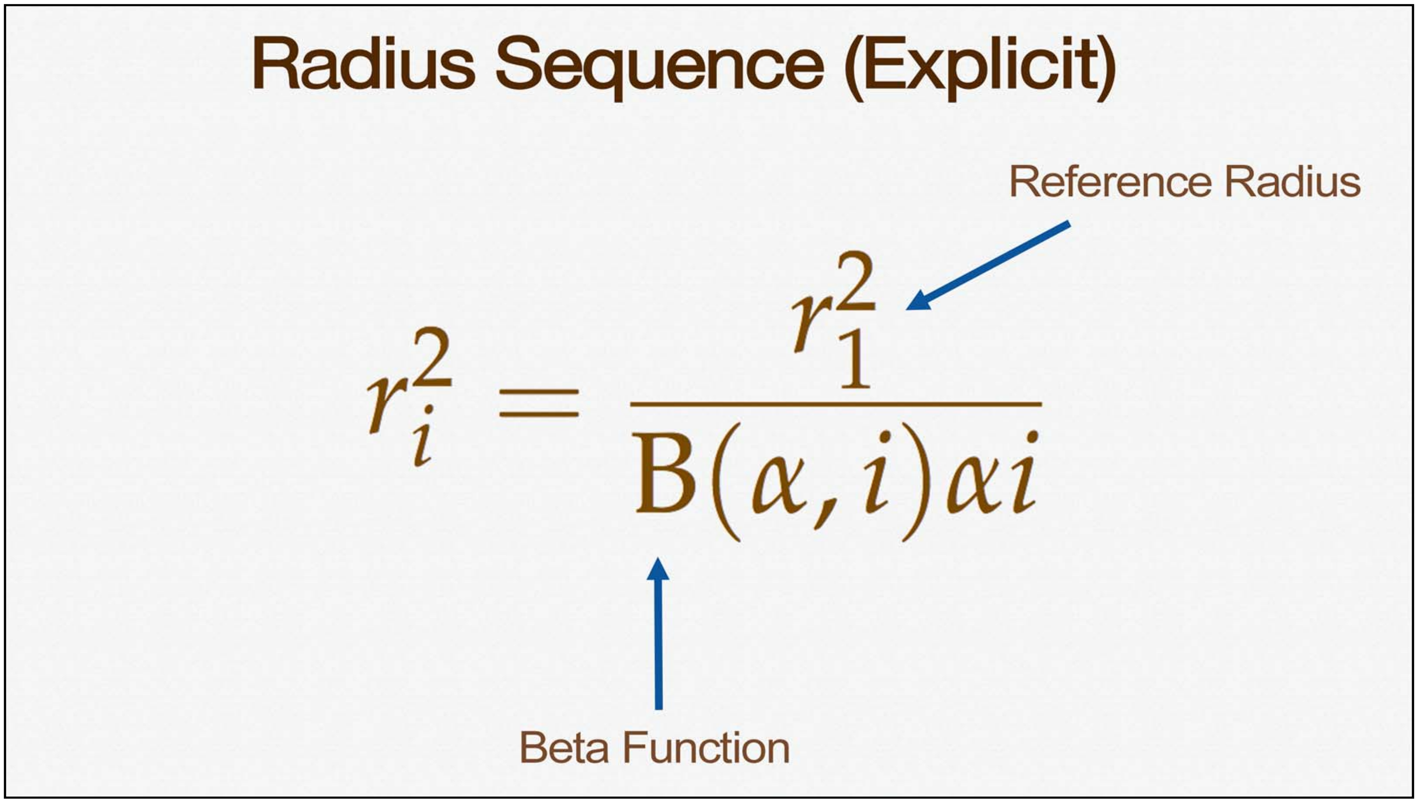
Local Statistics

Our Radius Sequence

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

No Local Statistics!

Our radius sequence is similar to the radius update rule proposed by Hachisuka et al., but with the important difference that it is entirely independent of local statistics such as collected number of photons, or local photon density. In fact, we show in the paper that, for locally constant photon density, the radius sequence is the same as the one from Hachisuka et al.



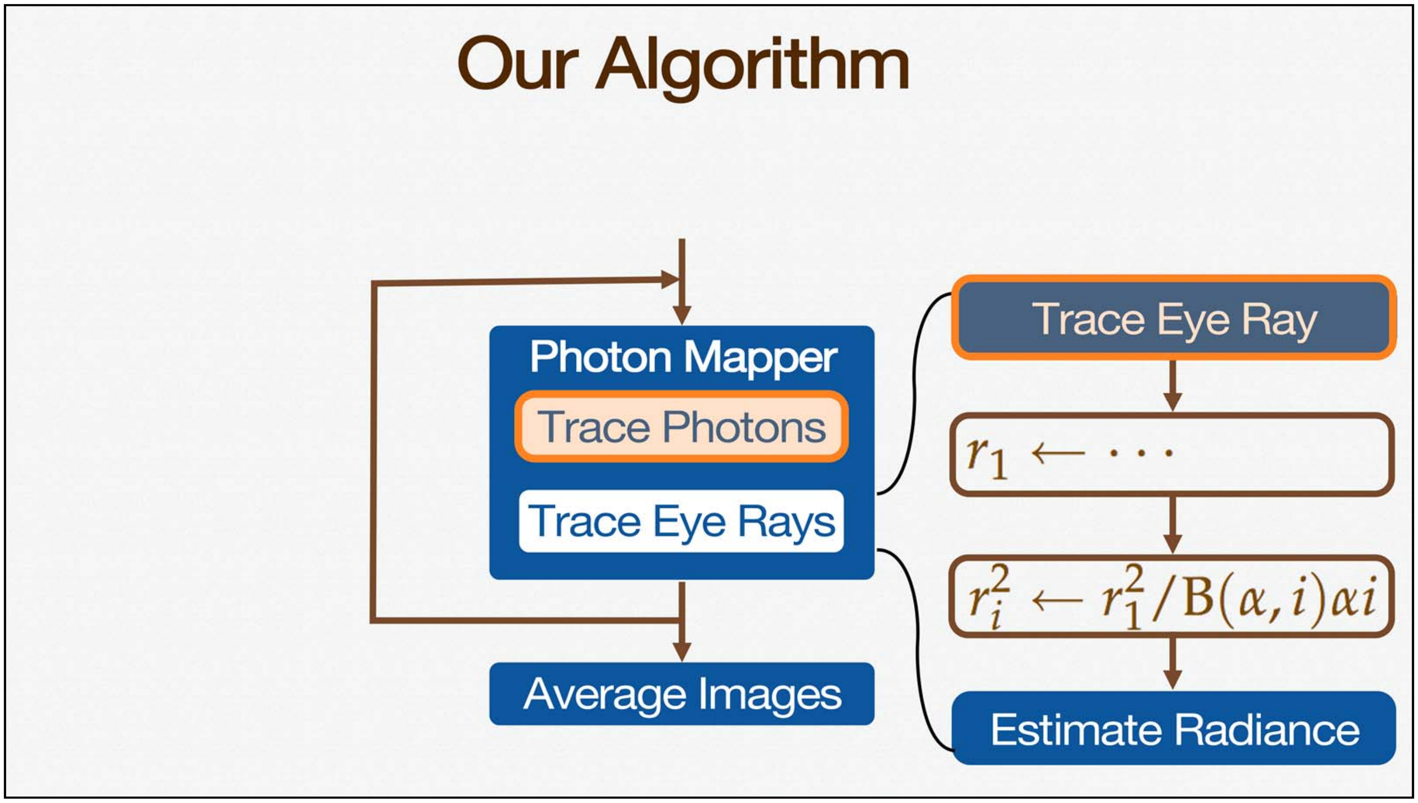
Radius Sequence (Explicit)

Reference Radius

$$r_i^2 = \frac{r_1^2}{\mathrm{B}(\alpha, i)\alpha i}$$

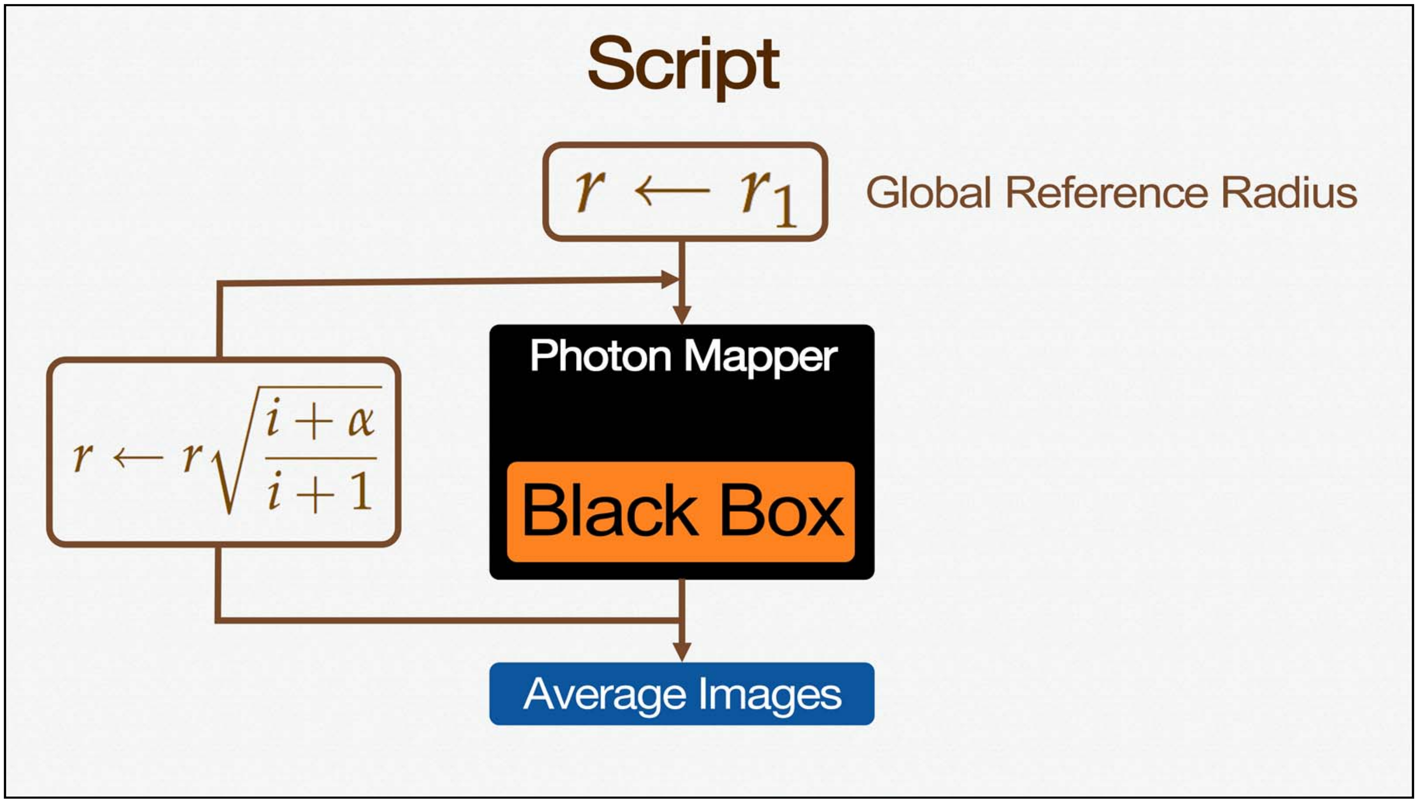
Beta Function

Instead of using a recursive formula, our radius sequence can also be written explicitly. r_1 is the initial reference radius which anchors the sequence. And B stands for the the Euler Beta function which is related the binomial coefficients.

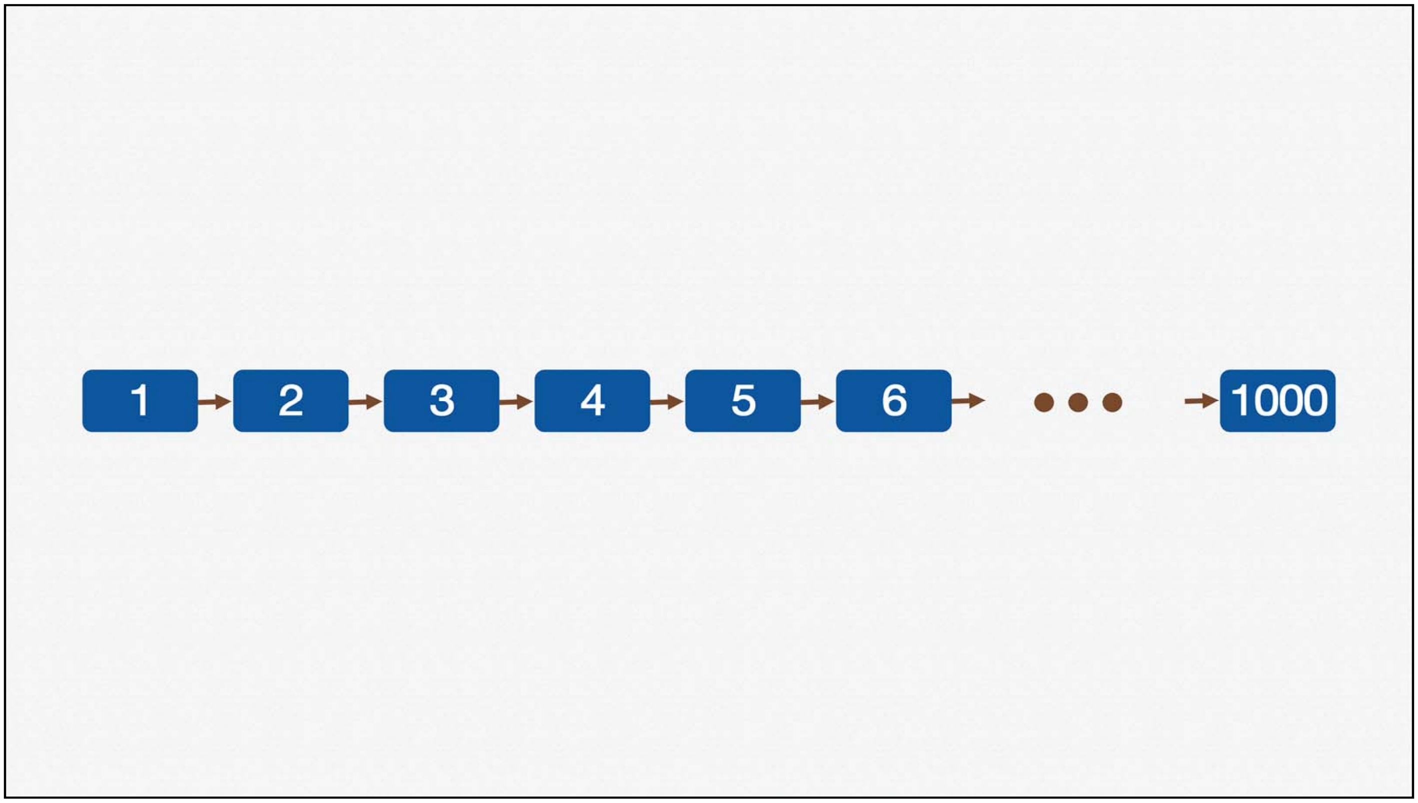


Our Algorithm

Using this formula, we propose the following algorithm. Our algorithm is a simple loop over a number of photon mapping iterations. The only specialty here is how we determine the kernel radius. In every iteration, just as in standard photon mapping, we perform the first pass by tracing photons and storing them in a photon map. Then in the second pass, for every traced eye ray, we determine a reference radius, and compute the radius of the current iteration using the explicit formula just shown before. This radius is then used to estimate the radiance with a range query.



There are different strategies to define the reference radius. The simplest solution is to define this reference radius globally. In this case, the implementation becomes much simpler; we can factor out the entire radius sequence and pass it as a parameter to the photon mapper, effectively treating it as a black box. Simultaneously, our progressive photon mapping algorithm collapses to a script.



In the original progressive photon mapping, the iterations were dependent on each other, because local statistics had to be carried over from one iteration to the next. Our iterations, on the other hand, are independent of any statistics and can therefore be executed in parallel.

1

2

3

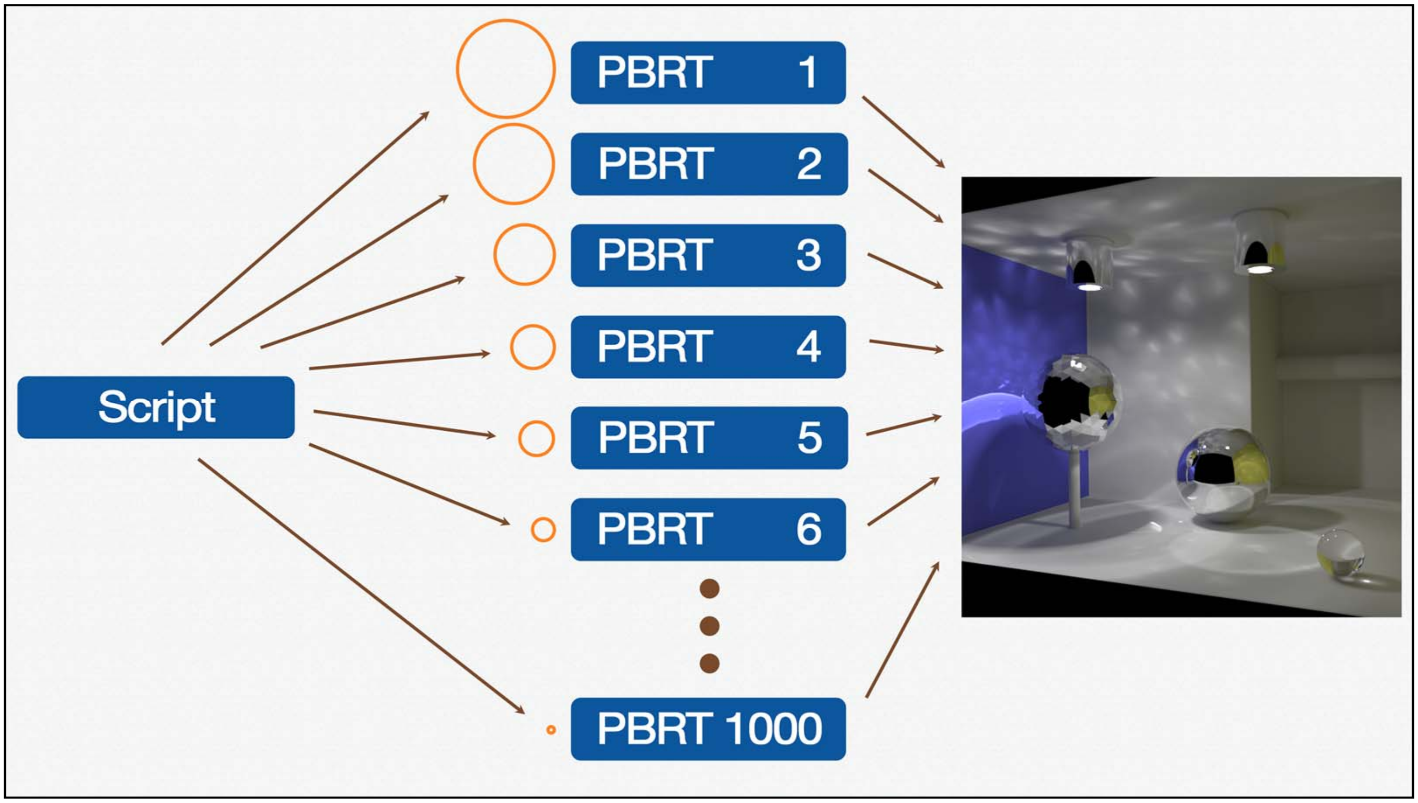
4

5

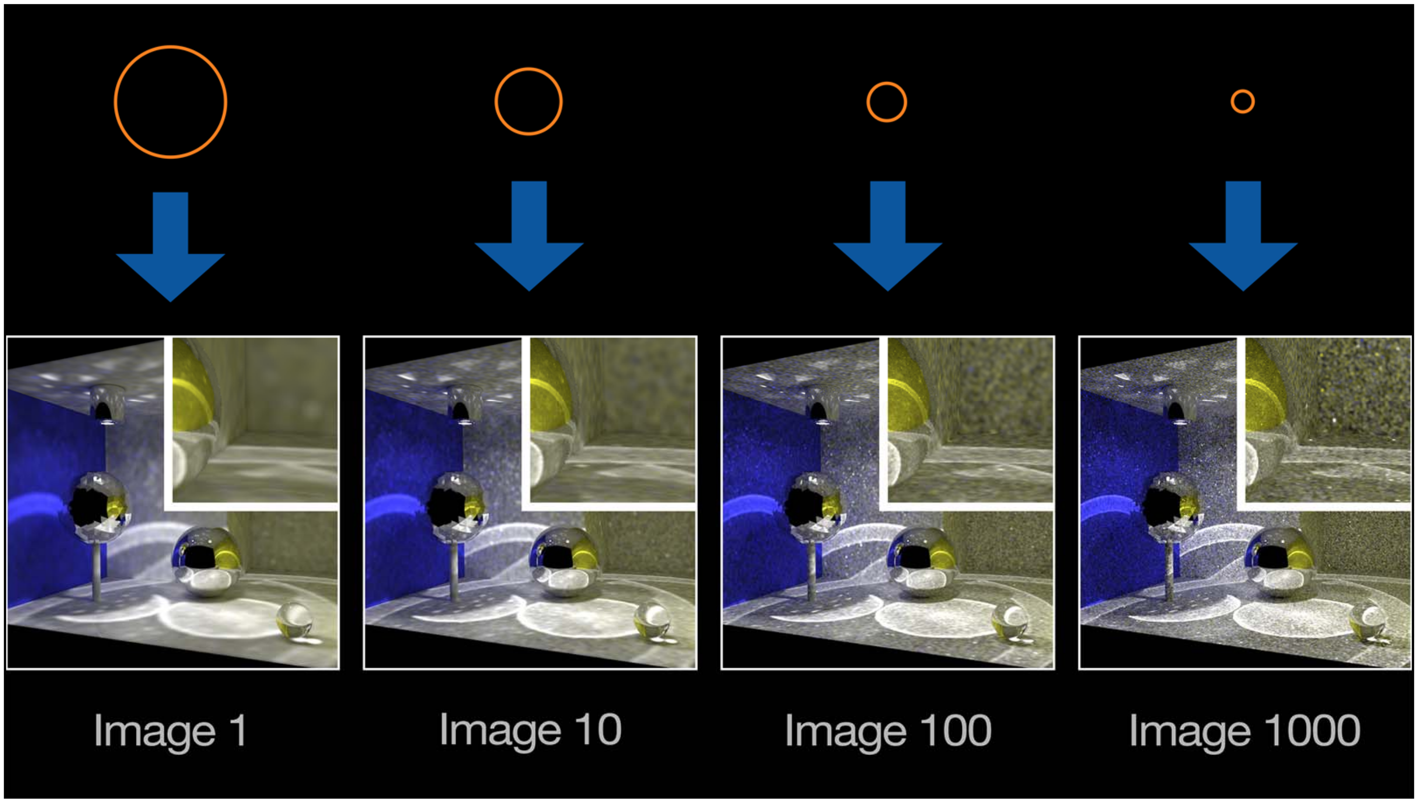
6

⋮

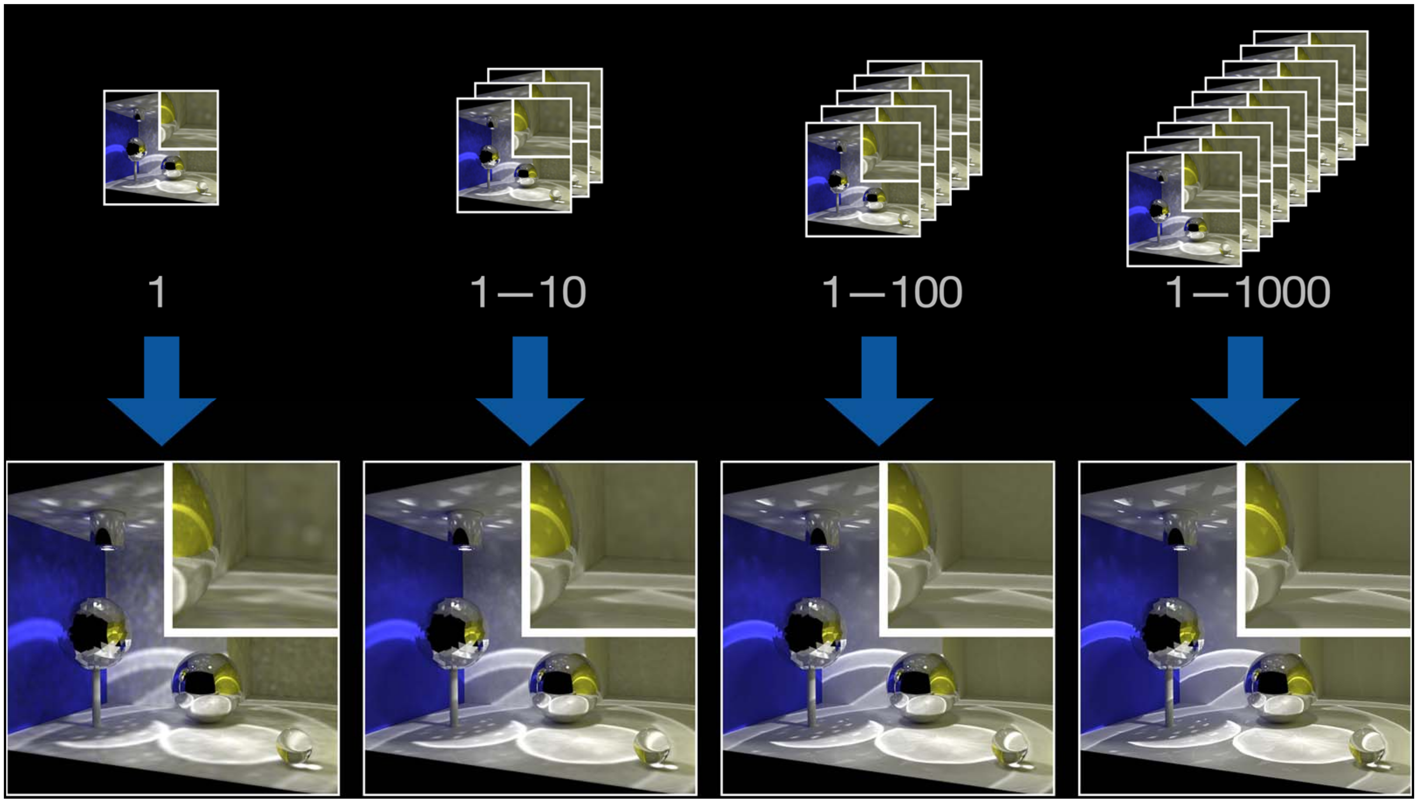
1000



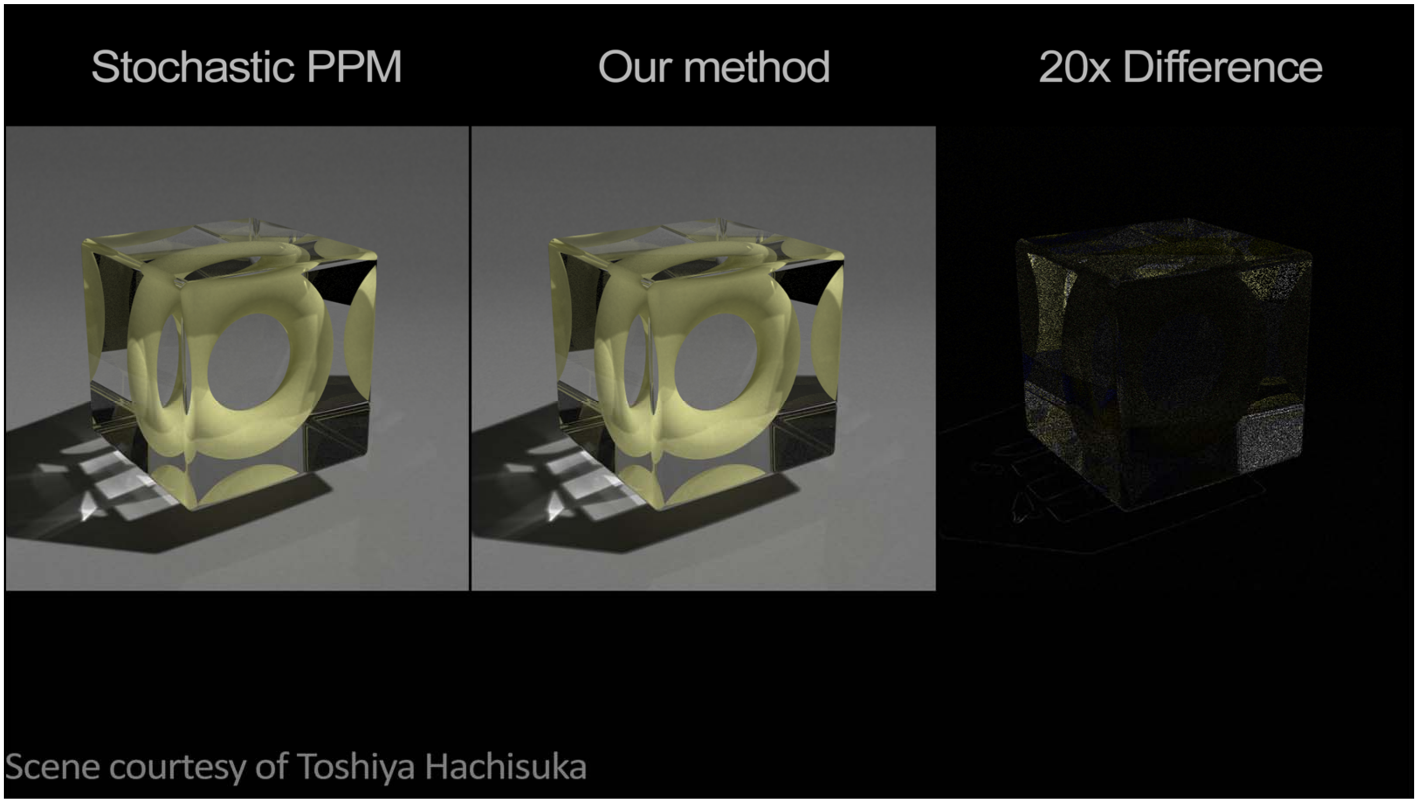
As a proof of concept, we have written a script to drive PBRT as a black box and to execute on a cluster. Every PBRT instance was fed with a radius from our radius sequence. The resulting images were then averaged on a single machine.



Here we see the results. As expected, the rendered images are noisier when the kernel radius is smaller.



But let’s see what happens when we average over the images. Here, we see just the first image. Now, the first 10 images averaged; already much less noise. Now, averaged over first 100 images. And finally, over 1000 images without visible noise or bias.



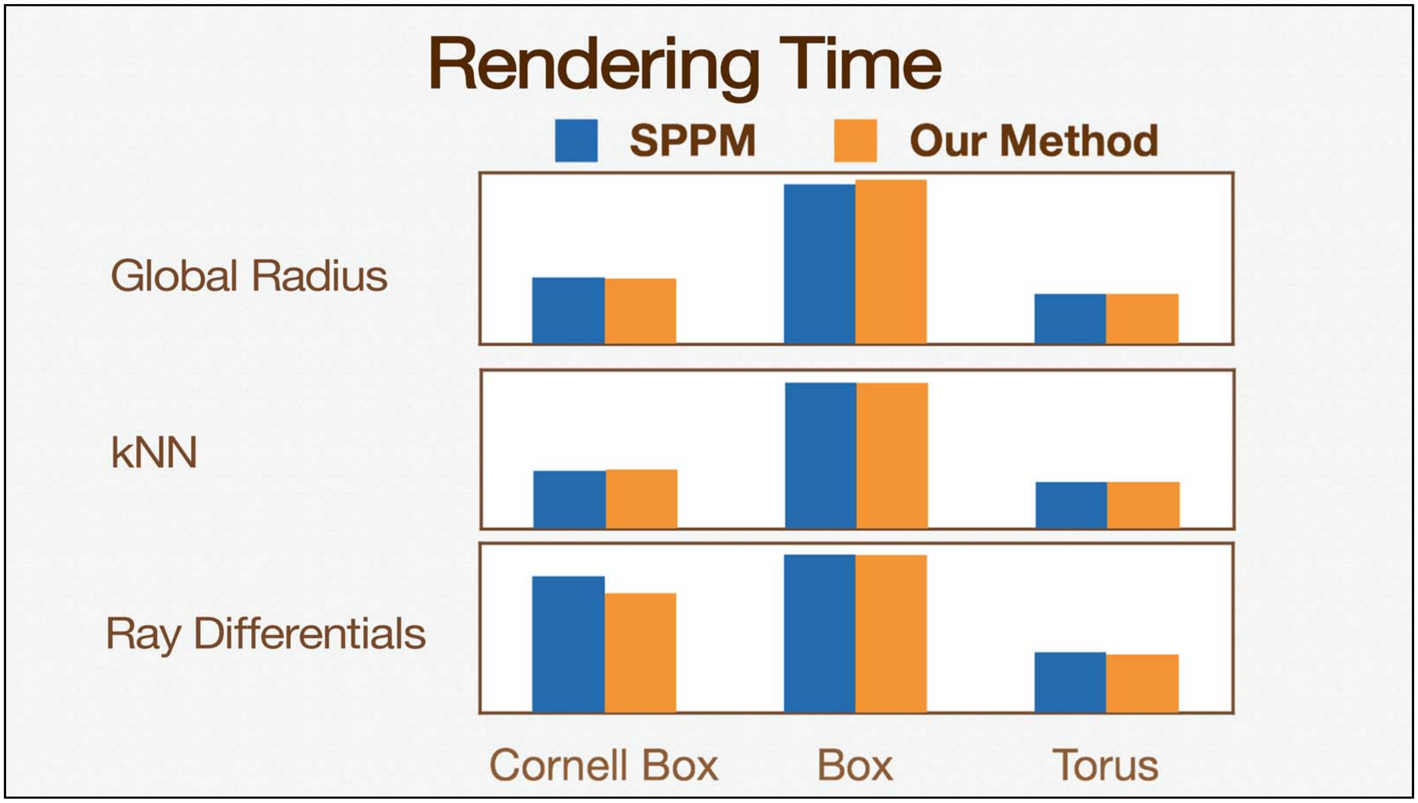
Stochastic PPM

Our method

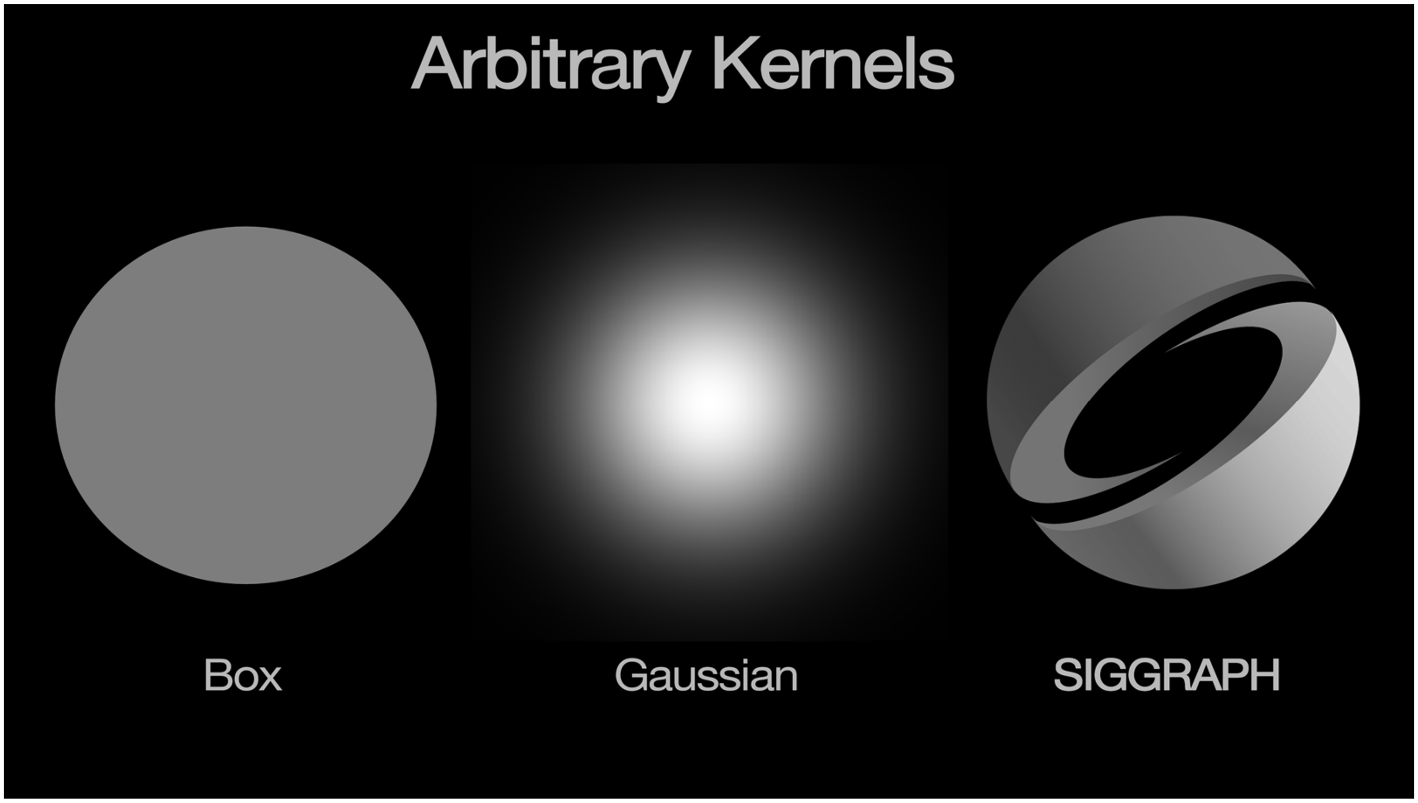
20x Difference

Scene courtesy of Toshiya Hachisuka

Quality-wise, our method stands on equal foot with traditional progressive photon mapping. Here we used a global reference radius for both methods. The difference between using traditional progressive photon mapping and our method is only in the noise.



Performance is also the same. We compare three reference radius strategies, global radius, k-nearest-neighbors, and ray differentials. In all cases, the difference in rendering time is negligible. This is not so surprising since the overhead for both methods is minimal.



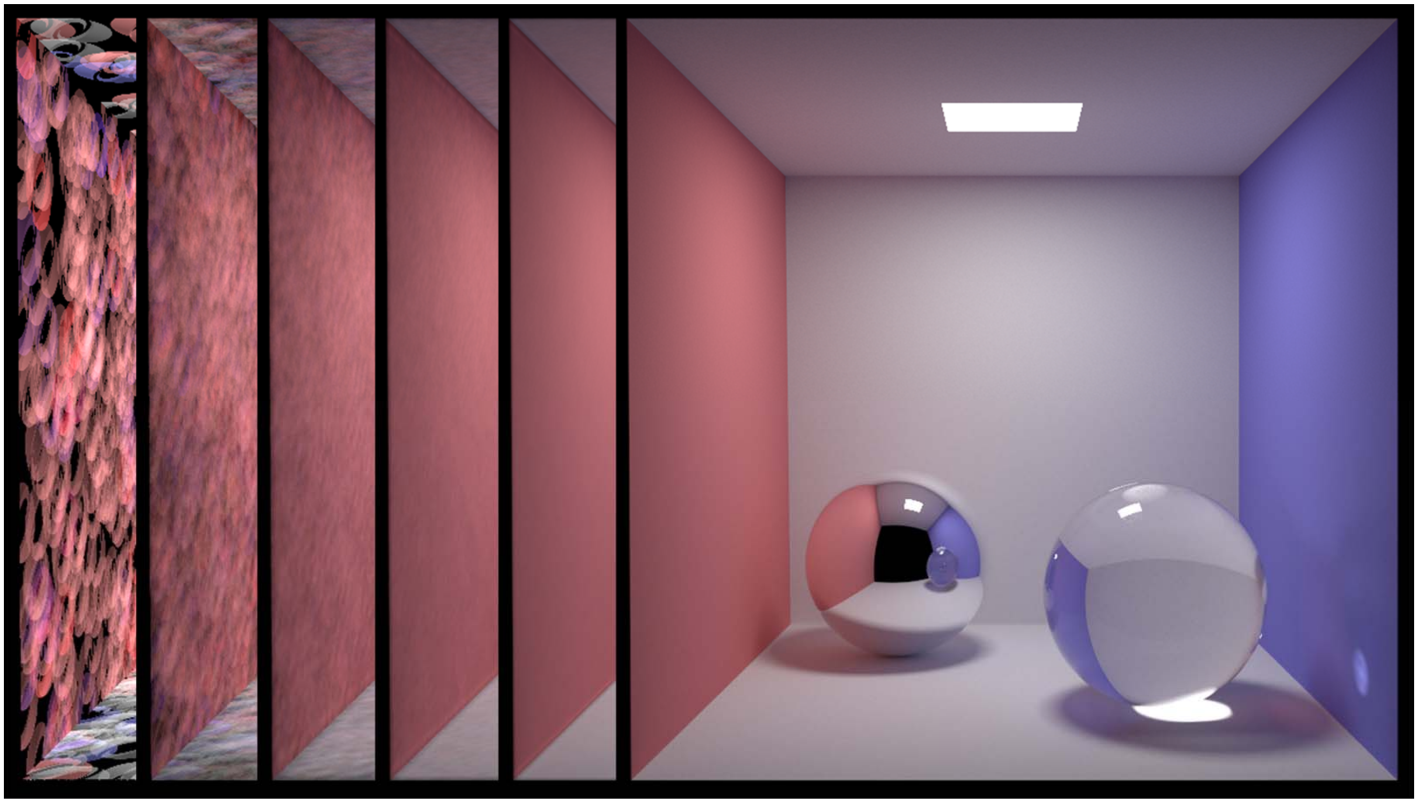
Arbitrary Kernels

Box

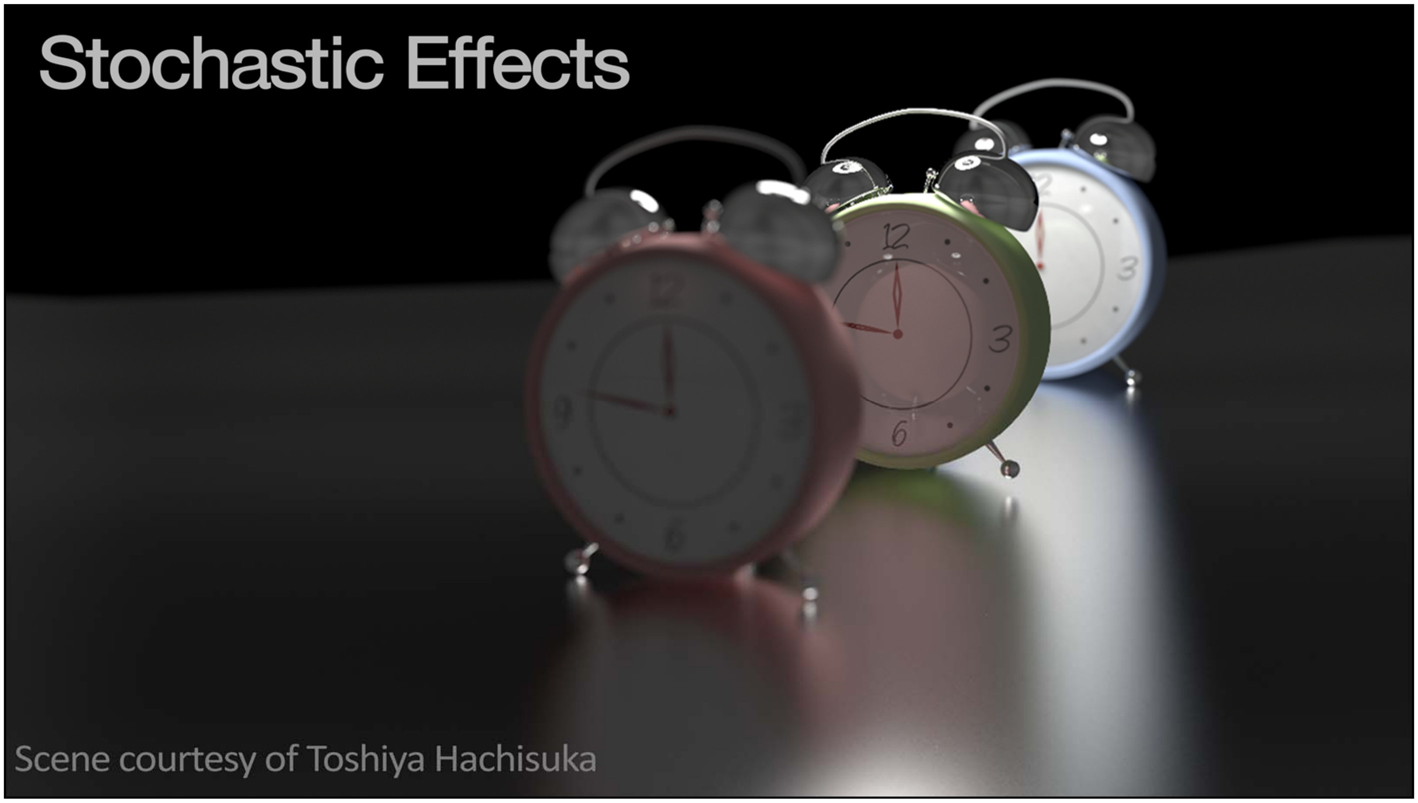
Gaussian

SIGGRAPH

Our reformulation of progressive photon mapping is more general. We can use arbitrary kernels for radiance estimation. Let’s see what happens if we use the right most kernel.



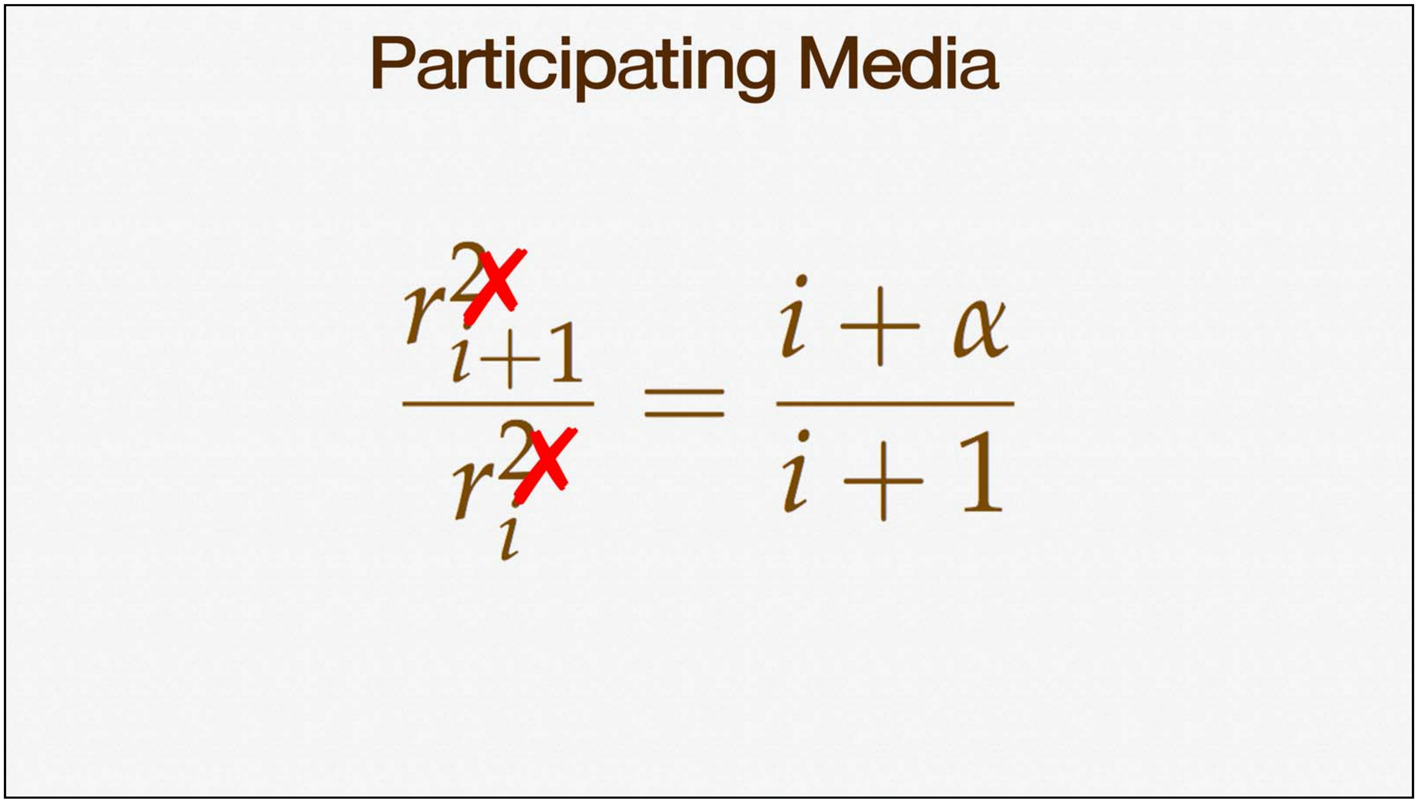
Here, we use only one thousand photons per iteration to show the kernel. Well, even in this case, after many iterations, we observe that the image sequence converges.



Stochastic Effects

Scene courtesy of Toshiya Hachisuka

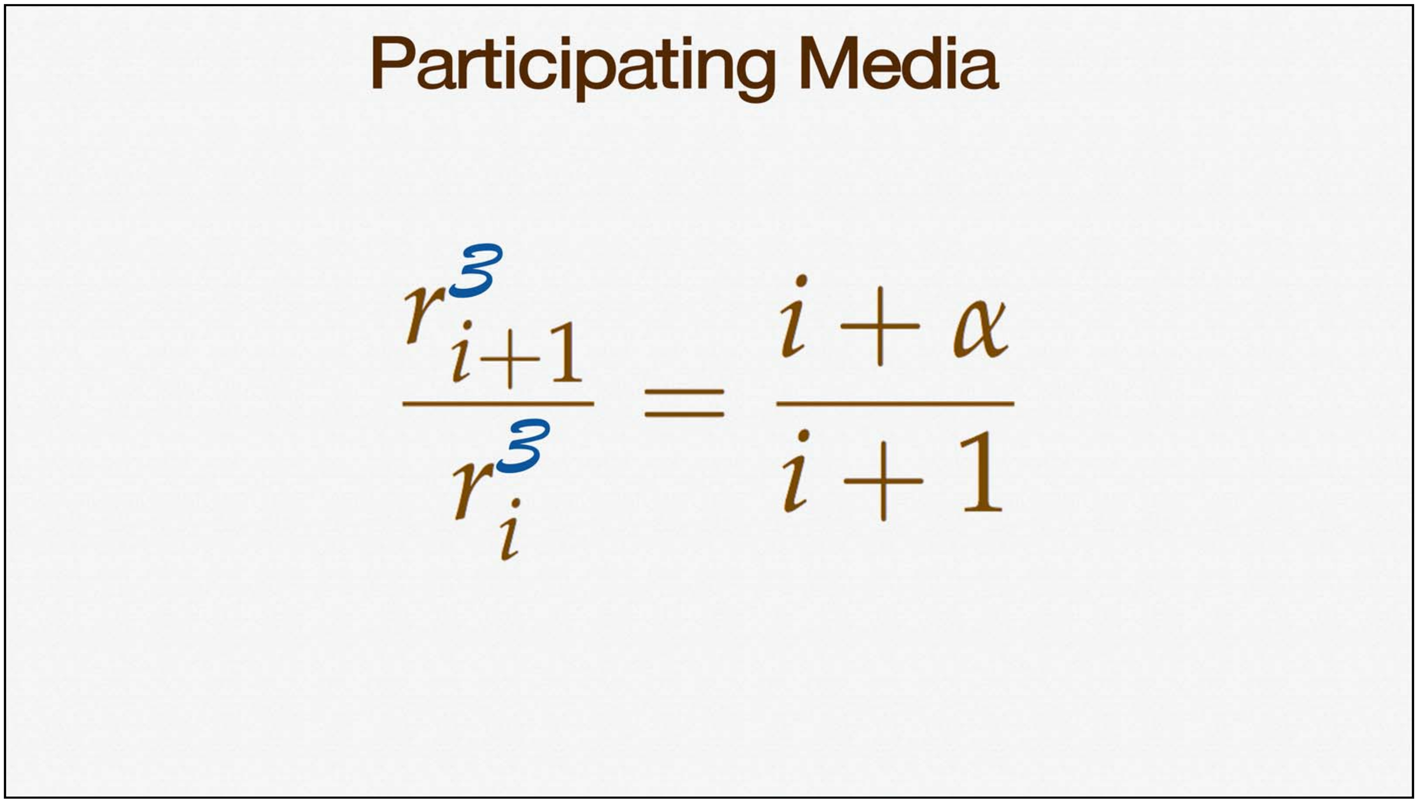
Just like stochastic PPM, our method includes stochastic effects as well. We demonstrate here depth of field and glossy surfaces.



Participating Media

$$\frac{r^{\xcancel{2}}_{i+1}}{r^{\xcancel{2}}_{i}} = \frac{i+\alpha}{i+1}$$

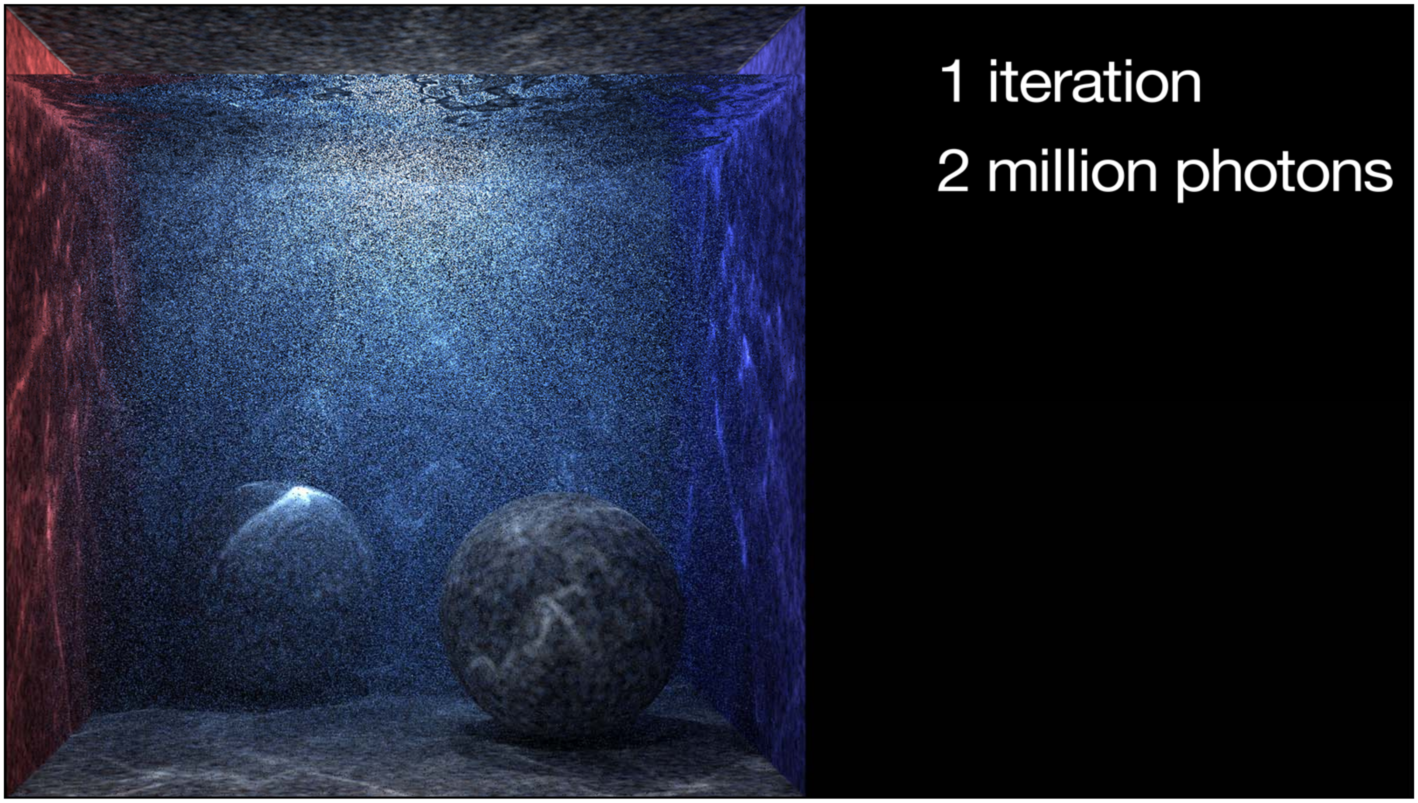
Last not least, our derivation of radius sequence and analysis extends to participating media as well. In volumetric photon mapping, the radiance is not defined on surfaces, but in volume. The kernel used for radiance estimate becomes therefore three dimensional.



Participating Media

$$\frac{r_{i+1}^{3}}{r_{i}^{3}} = \frac{i+\alpha}{i+1}$$

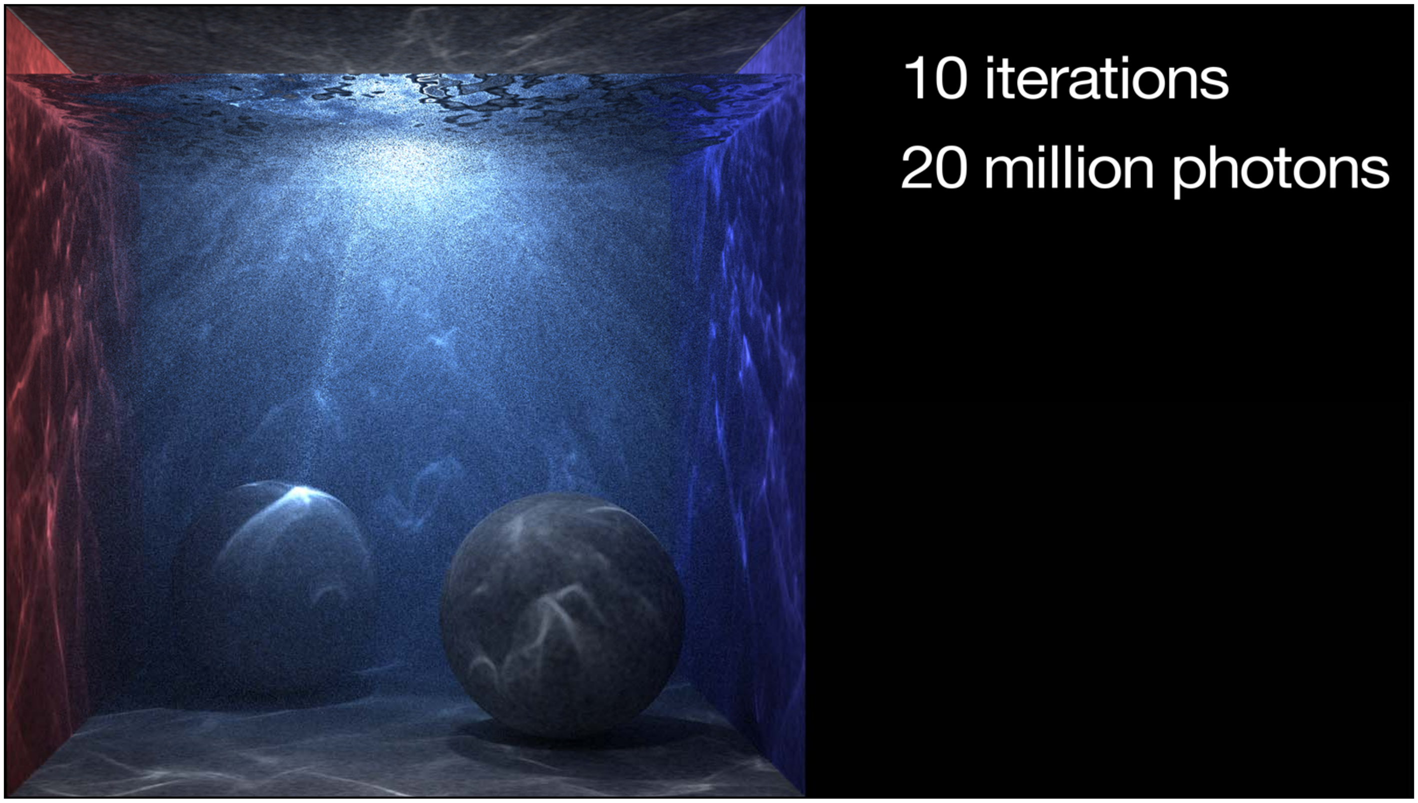
Following our analysis, we only have to replace in the radius sequence the exponents of the radii with 3, allowing us to integrate over a three dimensional domain.



1 iteration

2 million photons

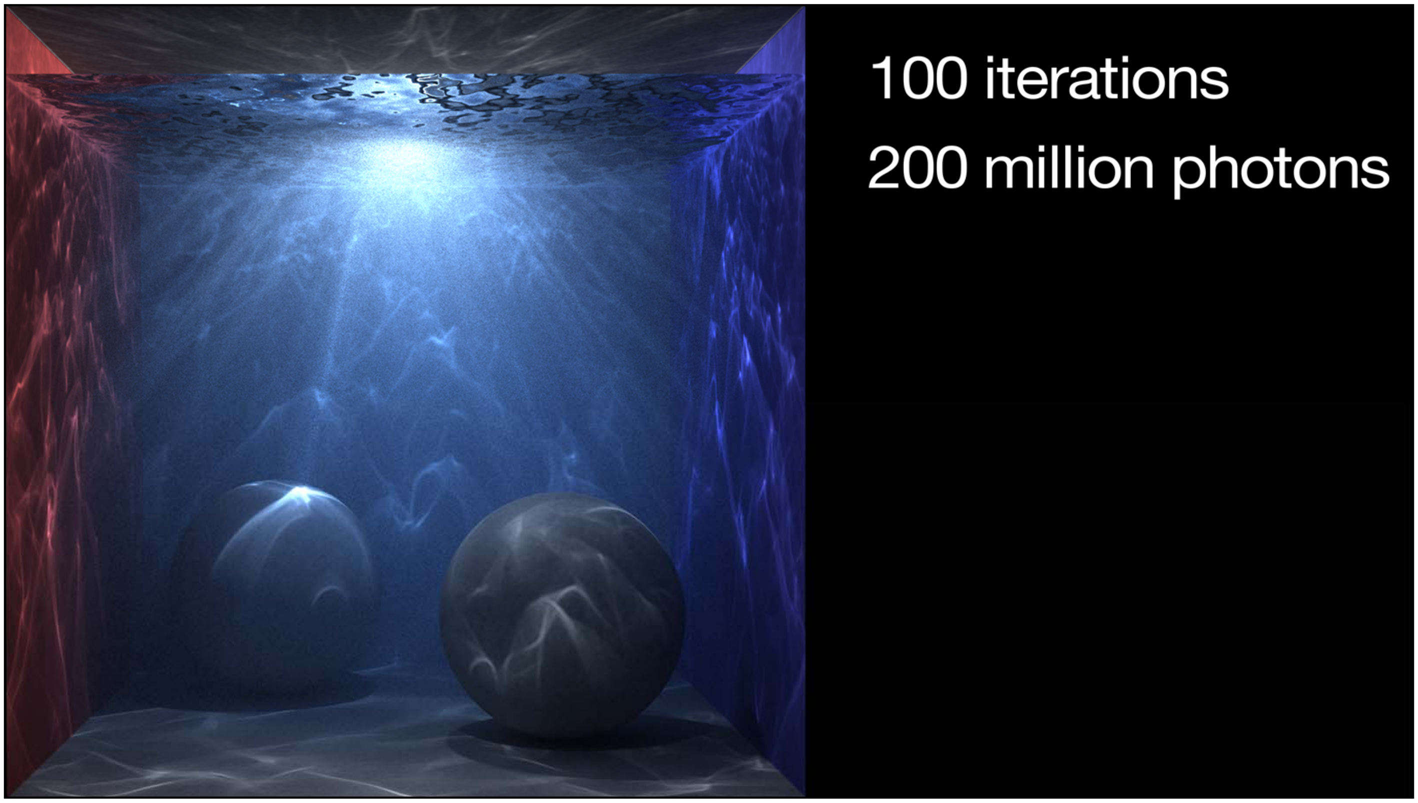
Here is a an example. This is a cornell box filled with water. At the beginning, with only 2 million photons, we only see noise.



10 iterations

20 million photons

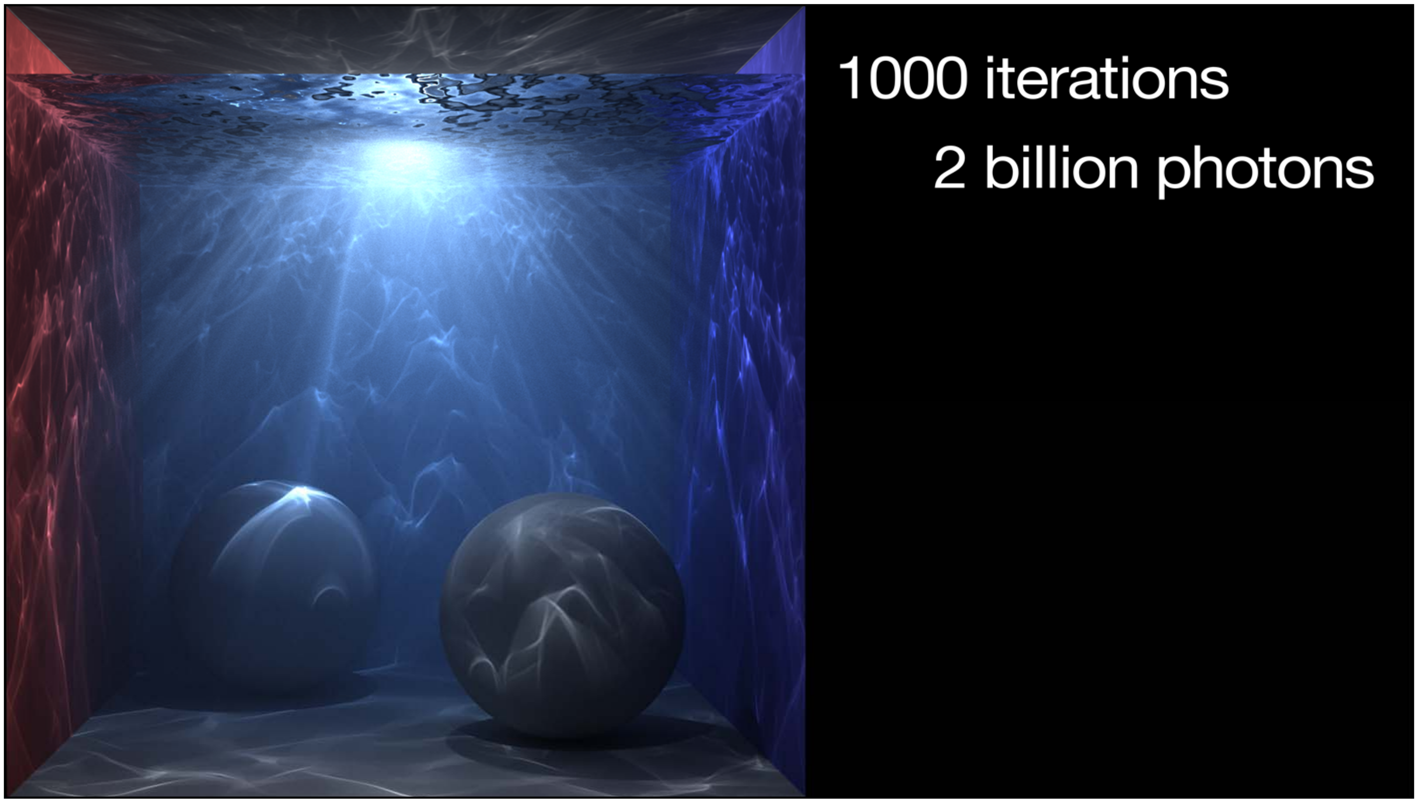
But if we shoot enough photons



100 iterations

200 million photons

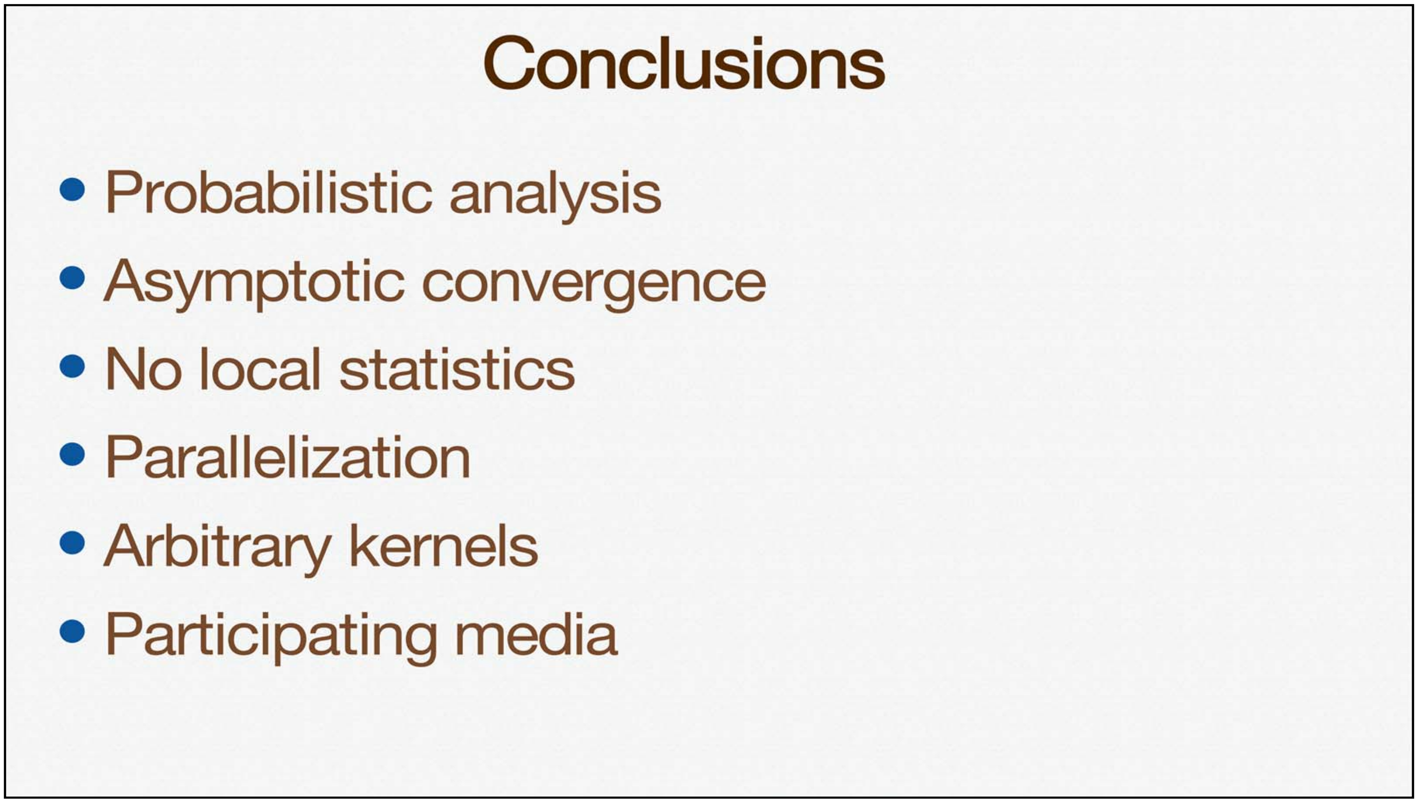
we start seeing god rays.



1000 iterations

2 billion photons

And eventually, after 1000 iterations and 2 billion photons, sharp caustics appear and the noise is gone.



Conclusions

- Probabilistic analysis
- Asymptotic convergence
- No local statistics
- Parallelization
- Arbitrary kernels
- Participating media

Let me review our results. We provided a probabilistic analysis and asymptotic convergence of progressive photon mapping. Most importantly, it leads to simpler implementations, which in the simplest case amounts to writing a script and using a standard photon mapper as a black box. Our method has the advantage of being parallelizable allowing the use of clusters. Furthermore, our reformulation

generalizes progressive photon mapping to arbitrary kernels.
And finally, our method trivially extends to other radiance estimates like volumetric radiance estimates and beam radiance estimates.

Acknowledgments

- Reviewers
- Hachisuka et al.
- Swiss National Science Foundation

Lastly, I would like to thank the reviewers for providing valuable feedback. I would also like to mention that all this work would not have been possible without the preceding work by Hachisuka et al. who developed the ingenious progressive photon mapping. Thank you for your attention!