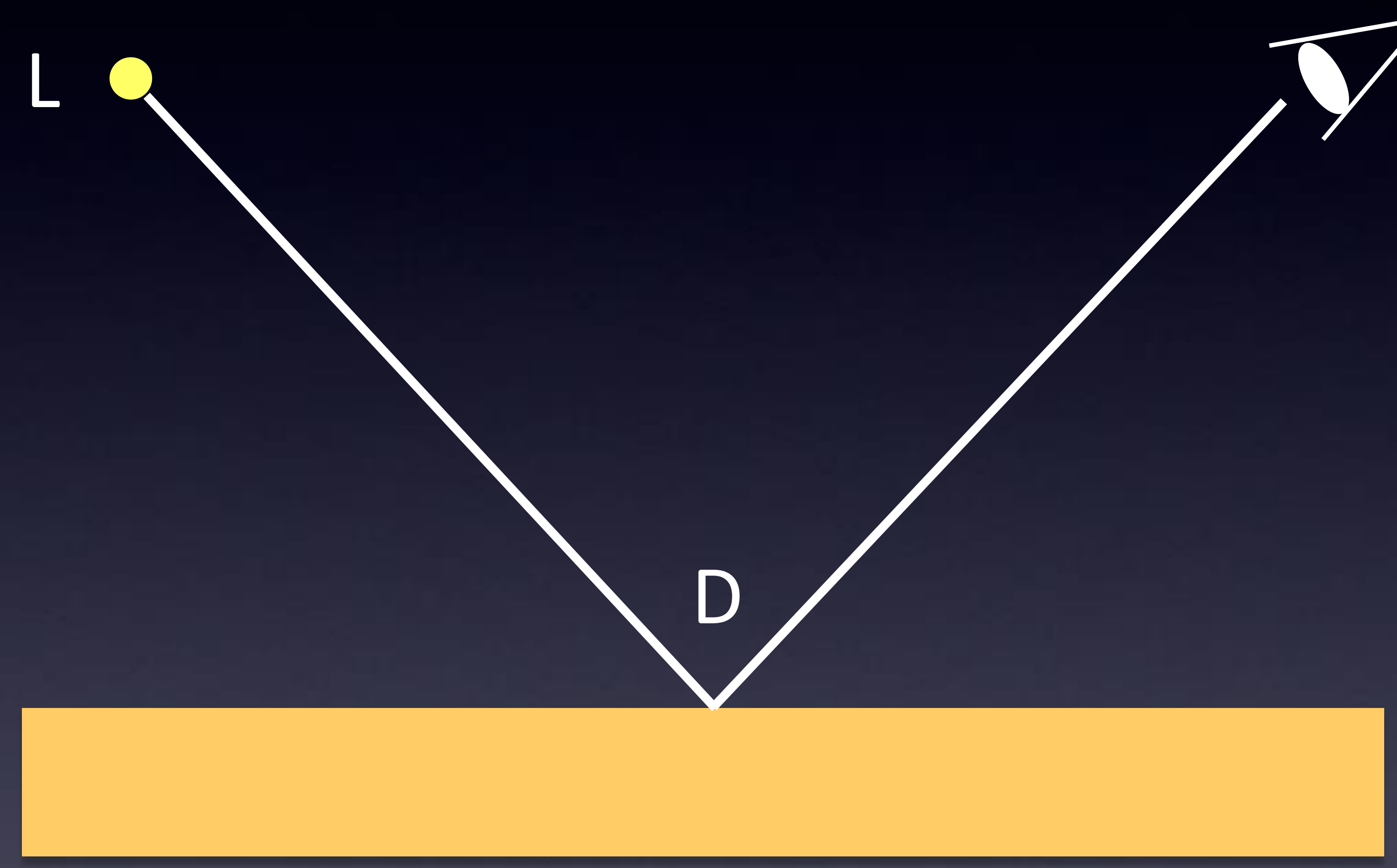
Progressive Photon Mapping

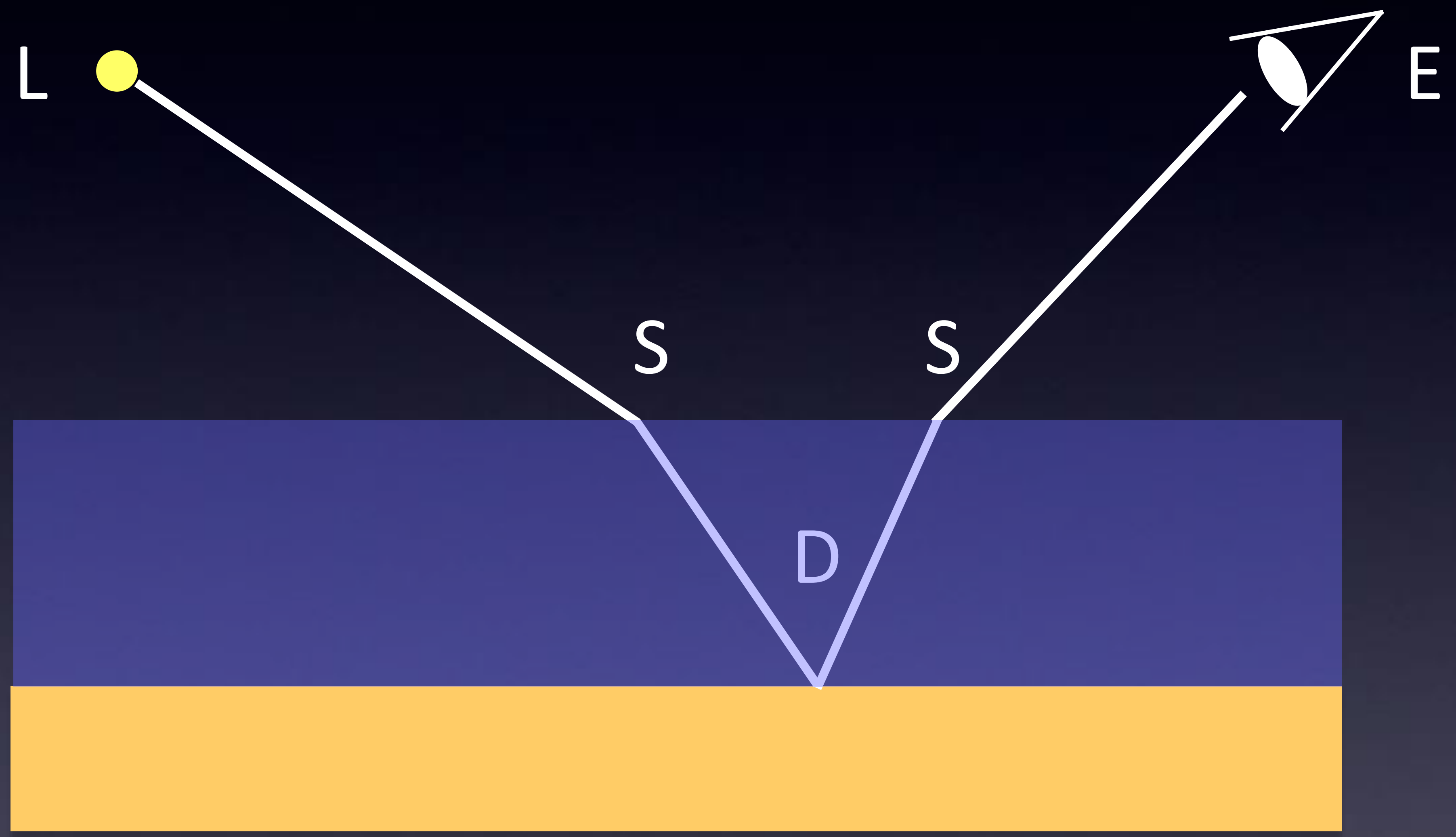
Toshiya Hachisuka[*] Shinji Ogaki[†] Henrik Wann Jensen[*]

[*]University of California, San Diego
[†]University of Nottingham

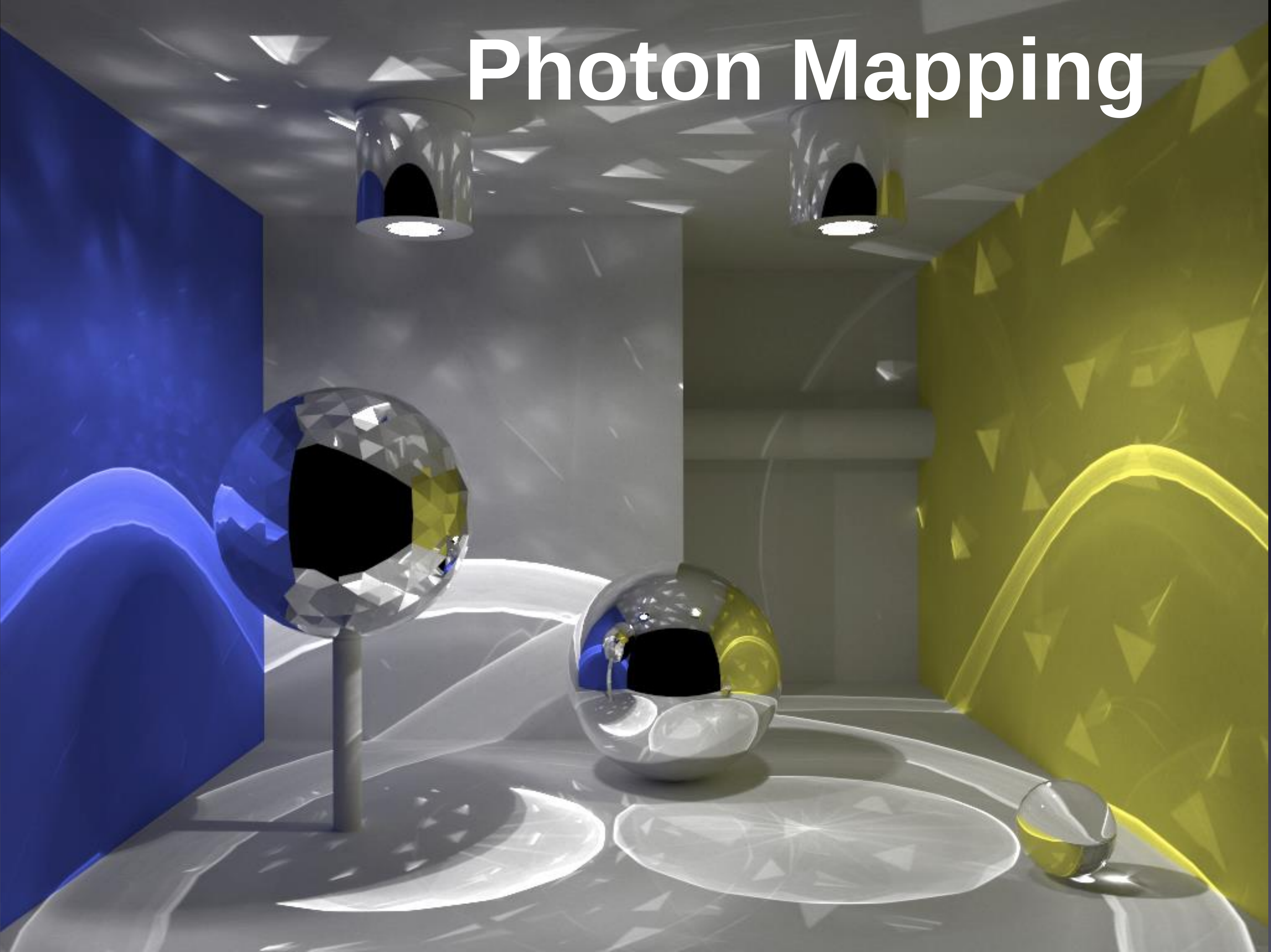
LDE Path



LSDSE Path

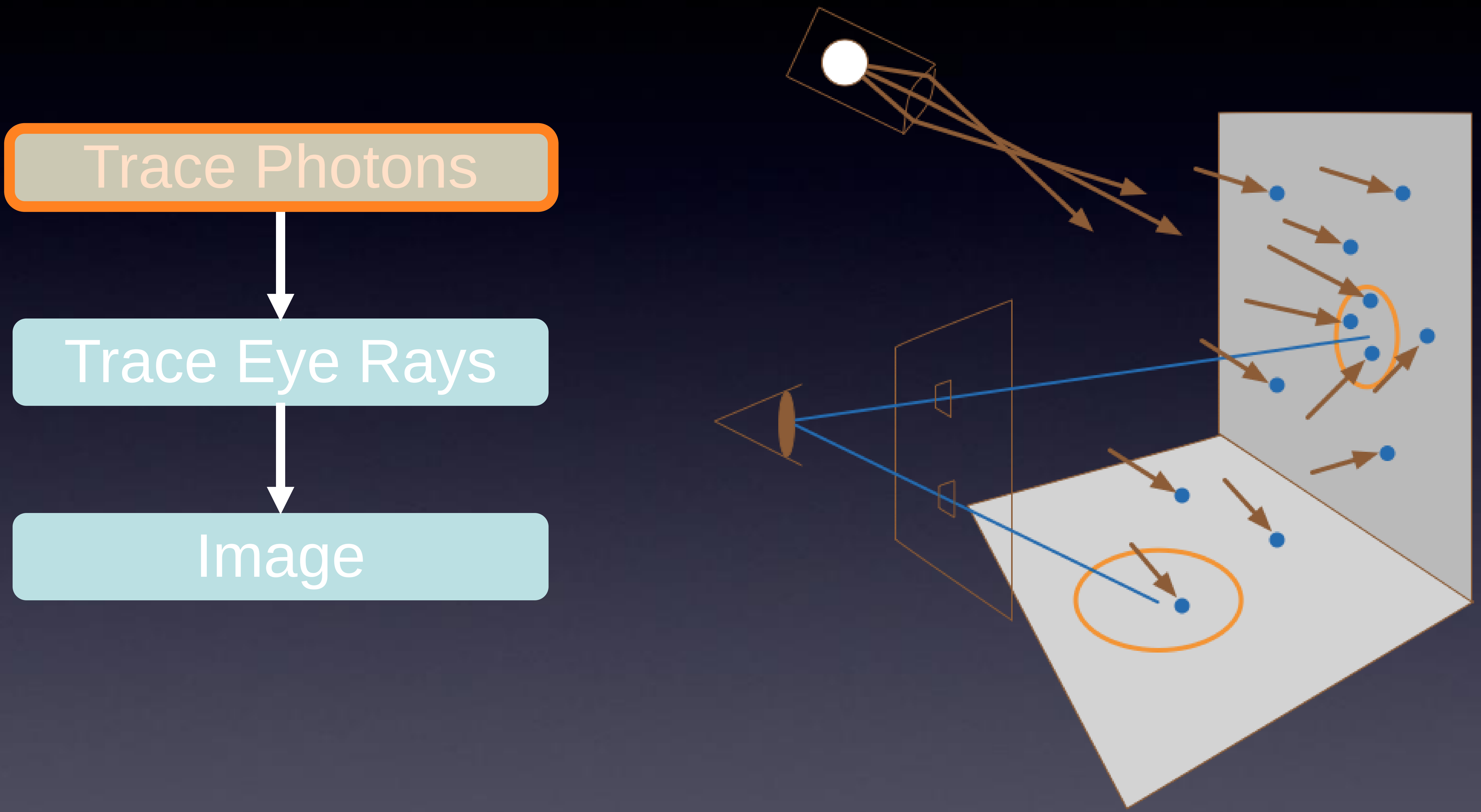



Path Tracing

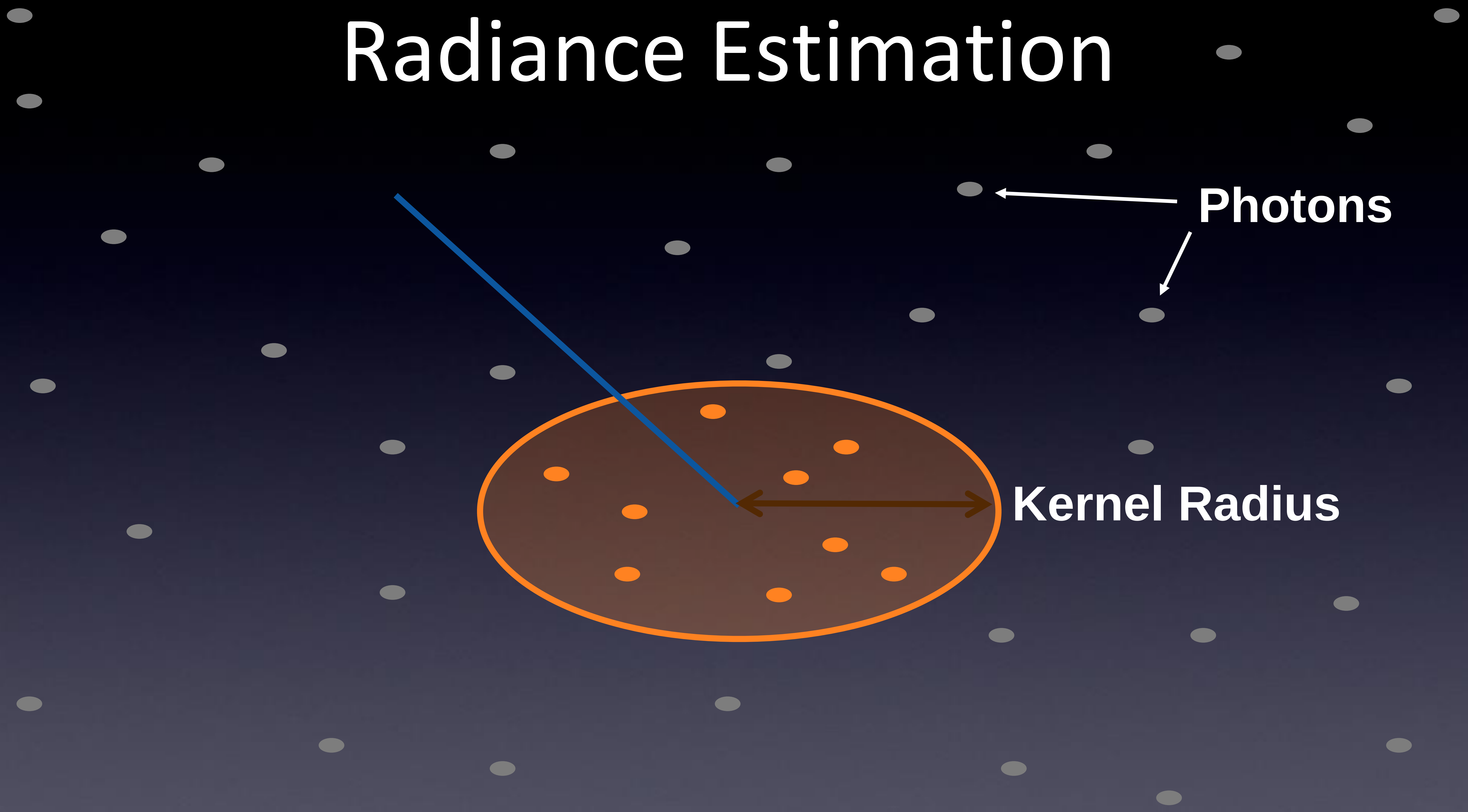


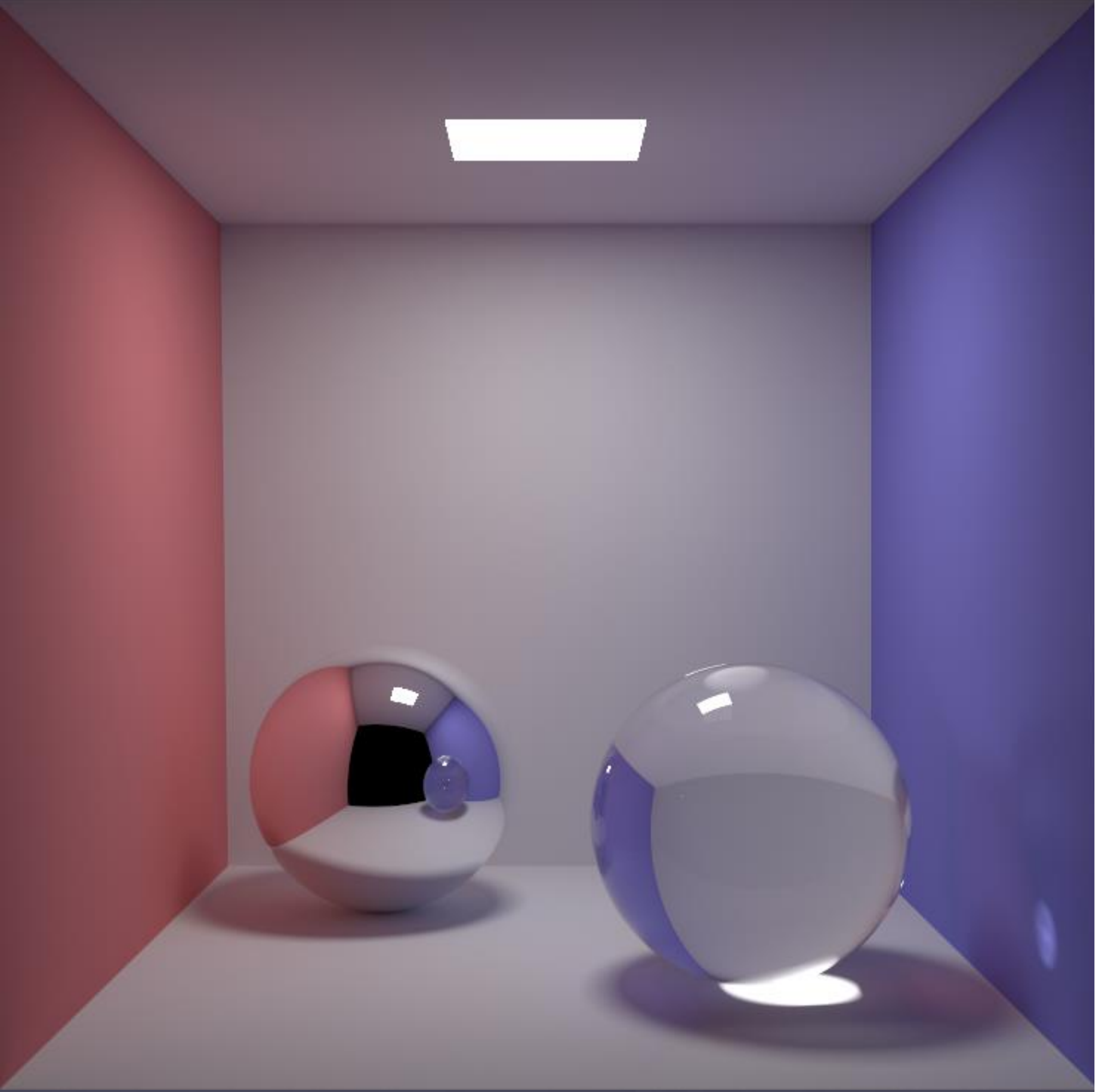
Scene courtesy of Toshiya Hachisuka

Photon Mapping

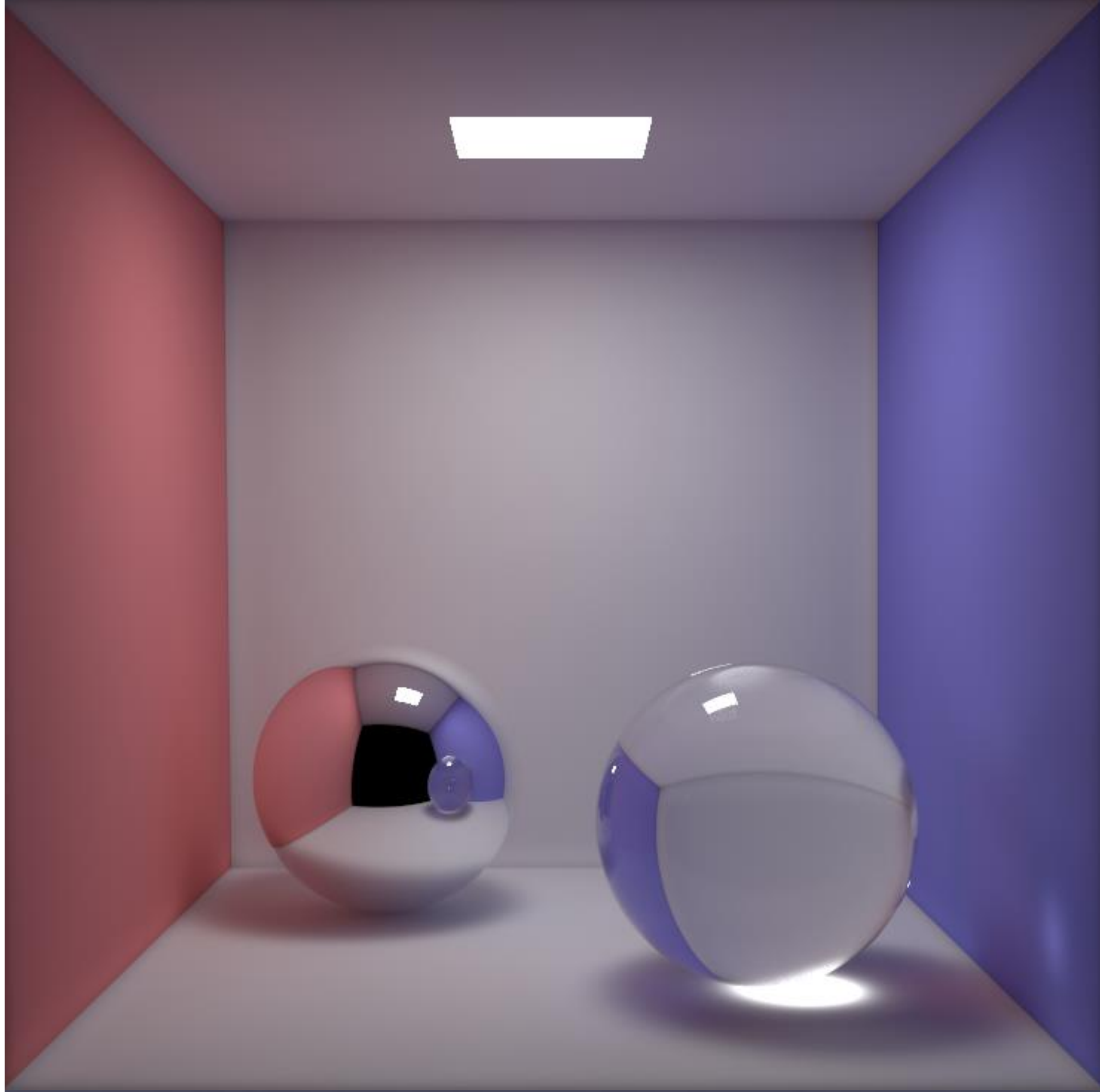


Radiance Estimation

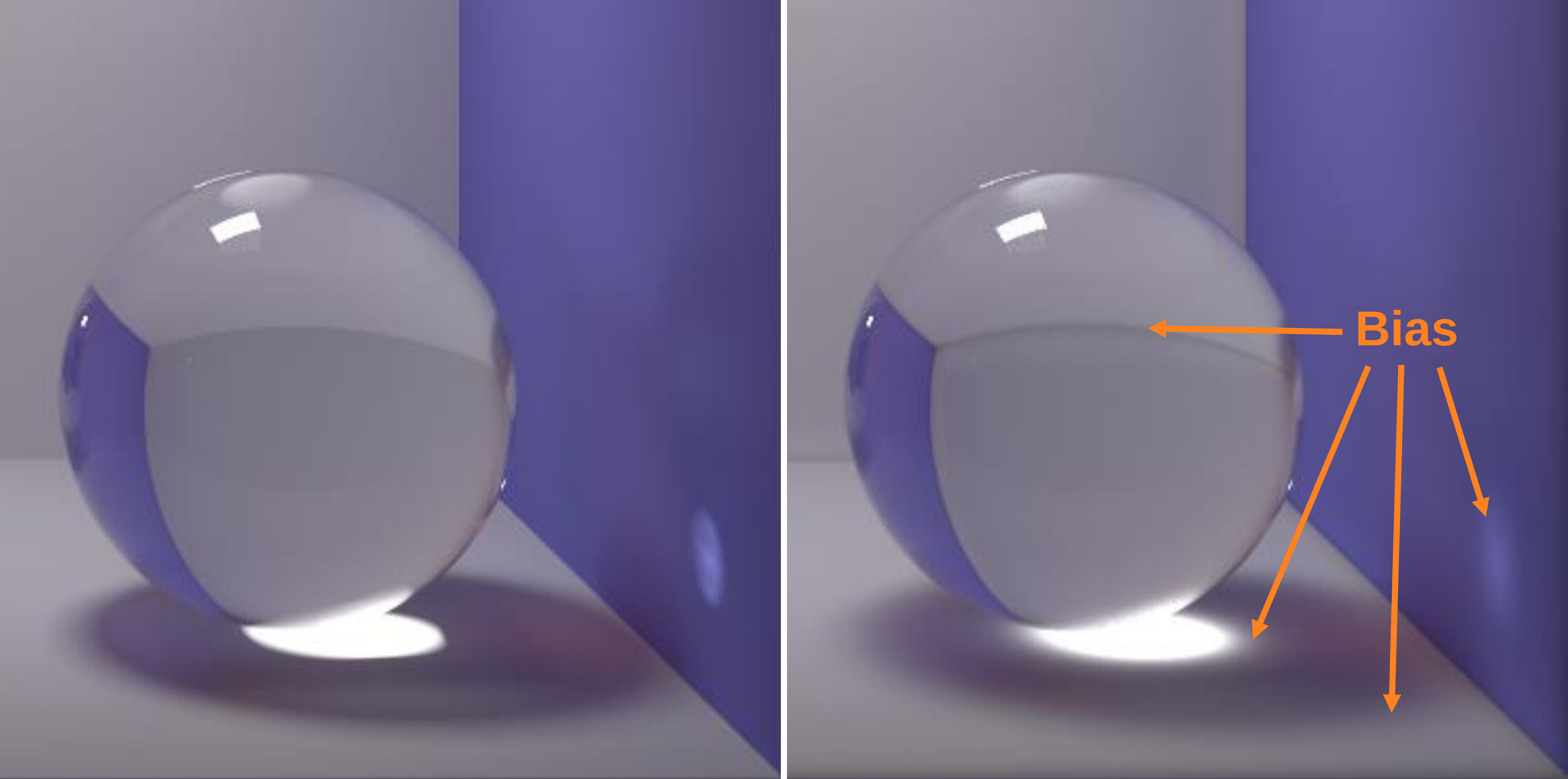




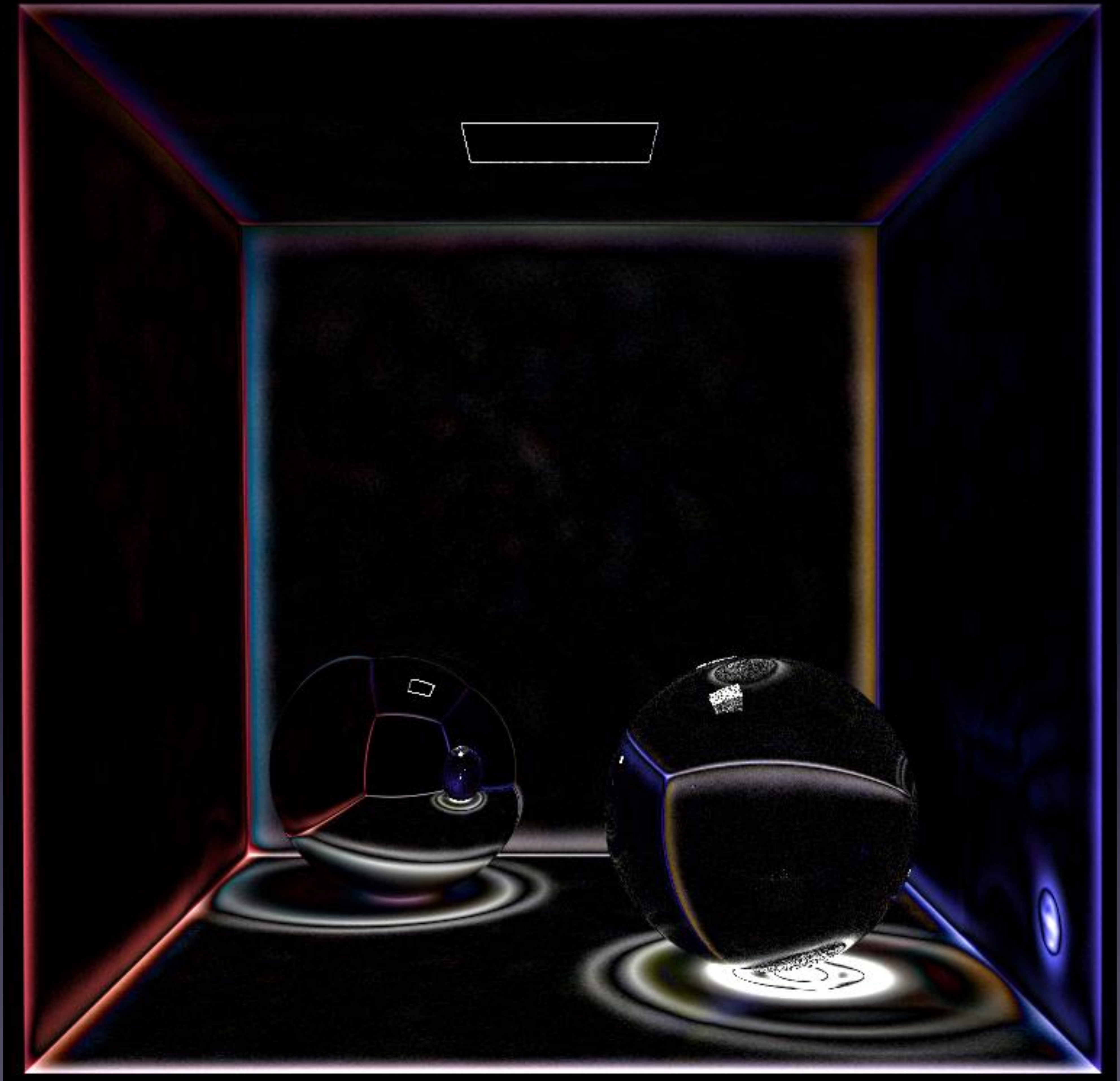
Ground Truth



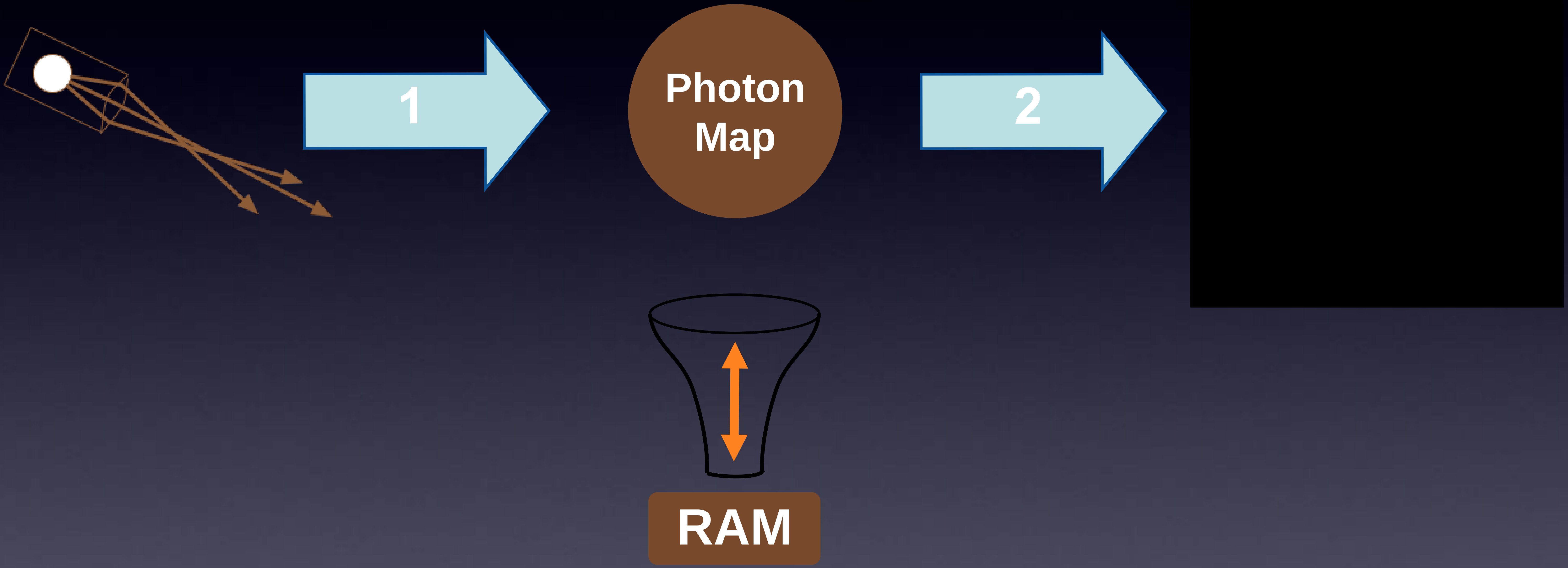
Photon Mapping



Ground Truth　　　Photon Mapping



Memory Bottleneck



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

Progressive Photon Mapping

First algorithm for computing *all* types of light transport with arbitrary accuracy

Progressive Photon Mapping

- New formulation of photon mapping
  - Robust for *any* light path including SDS path
  - Arbitrary accuracy using finite memory
  - New progressive radiance estimation algorithm
  - Easy to implement

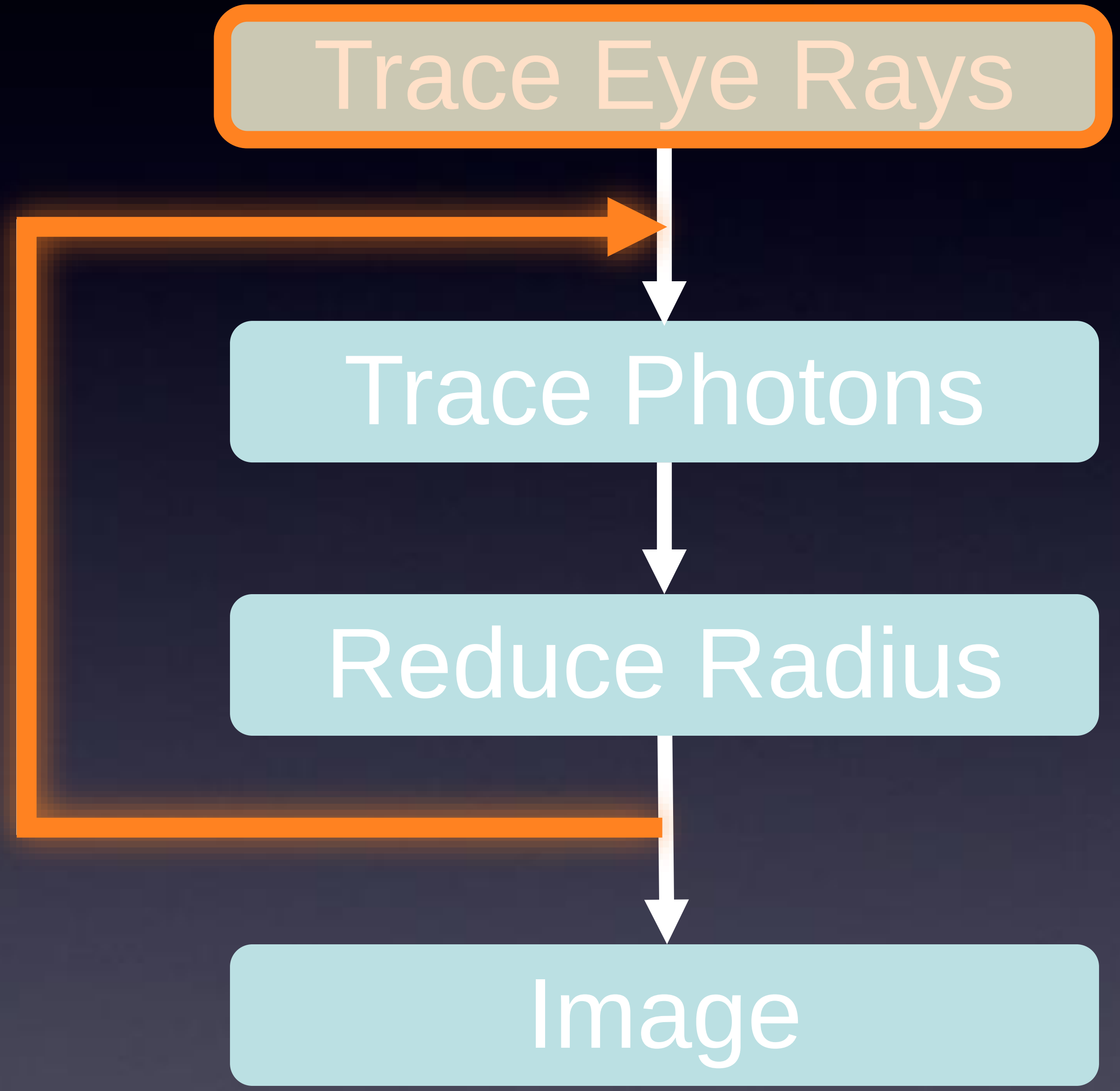
Progressive Photon Mapping

- Multi-pass method
  - Initial pass:
    points generation for radiance estimates
  - Refinement pass:
    - photon tracing
    - progressive radiance estimate

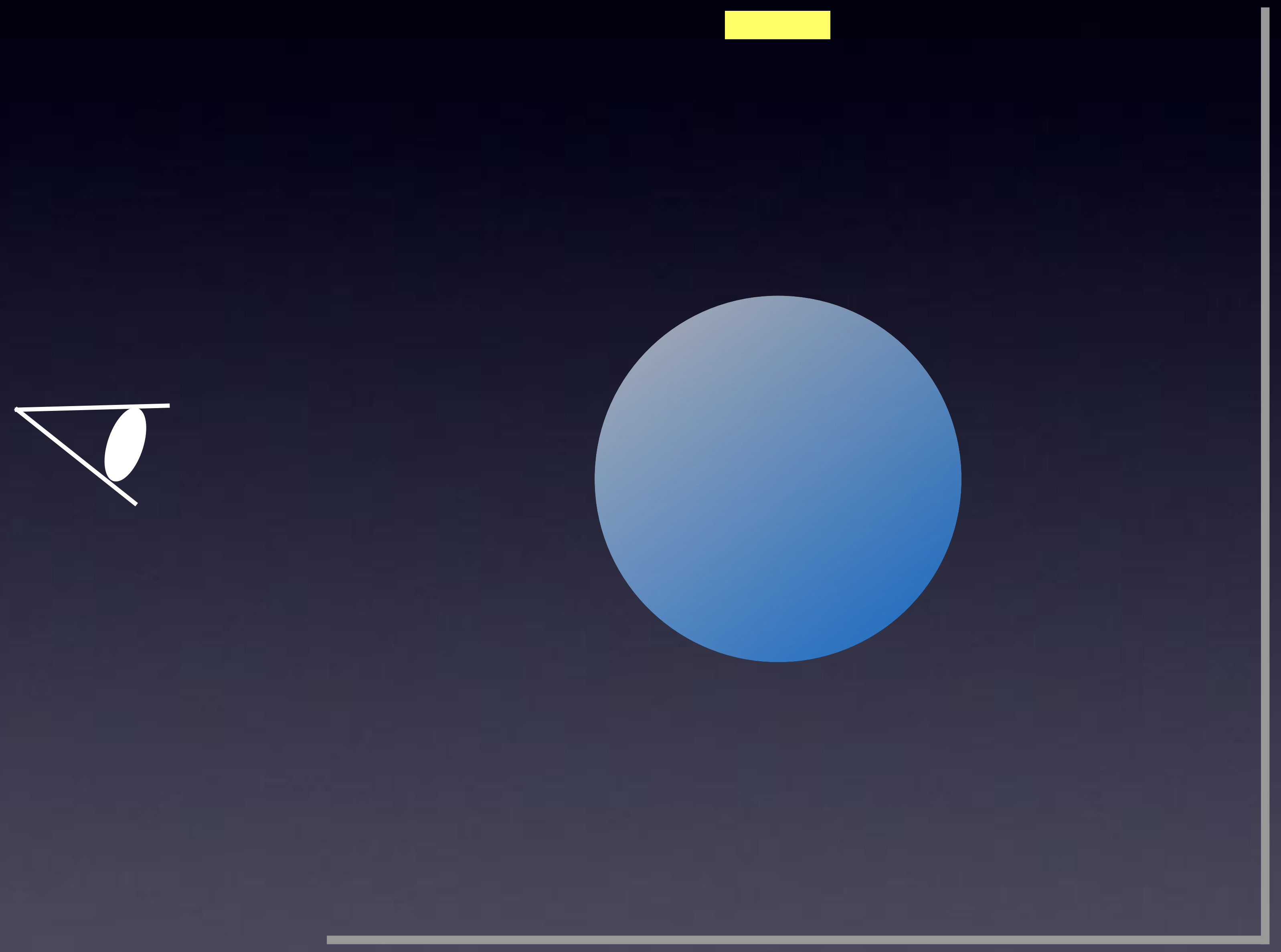
Key Idea

- Progressive radiance estimation
  - New density estimation algorithm
  - Converges to the correct value

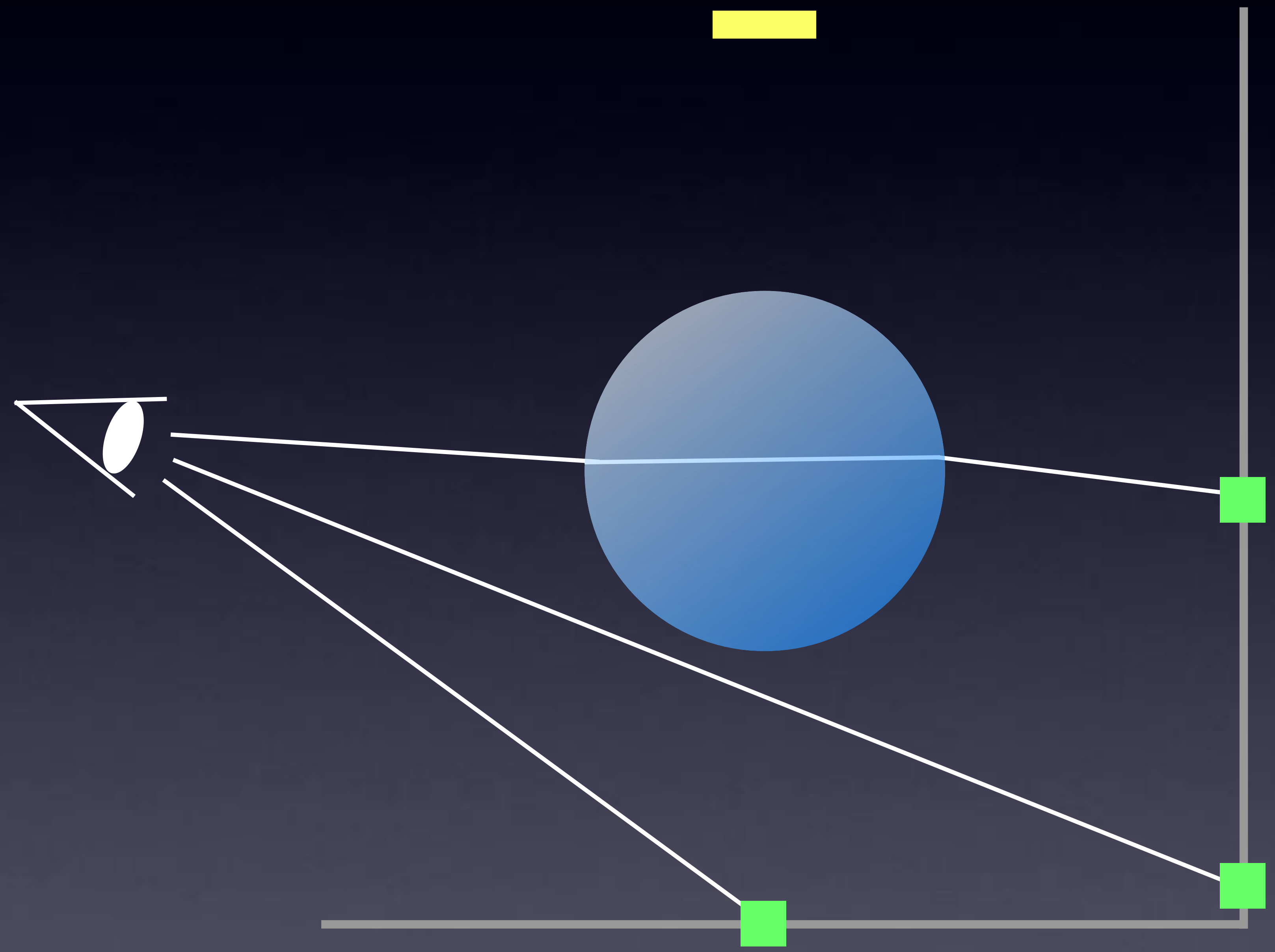
Progressive Photon Mapping



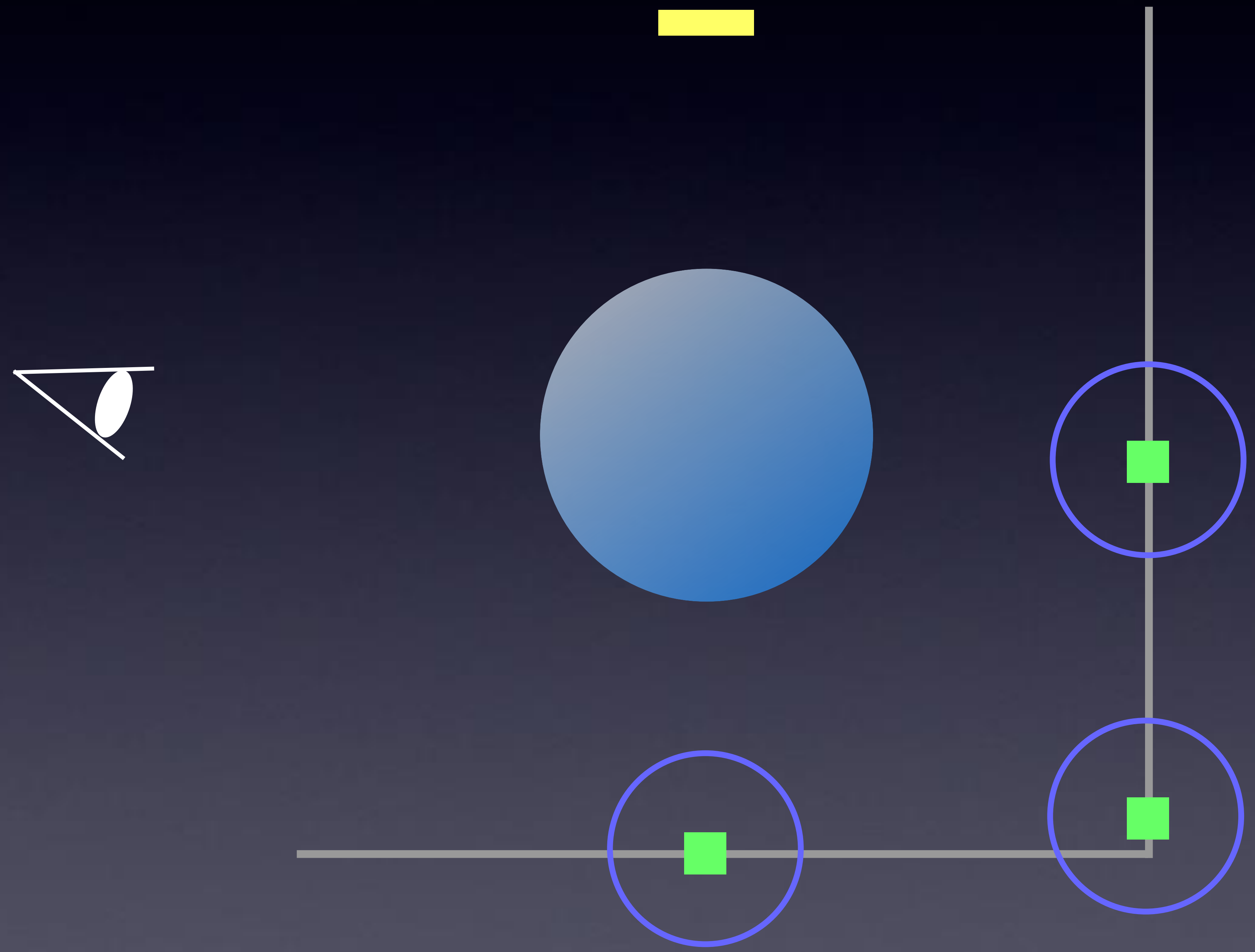
Progressive Photon Mapping - Initial Pass



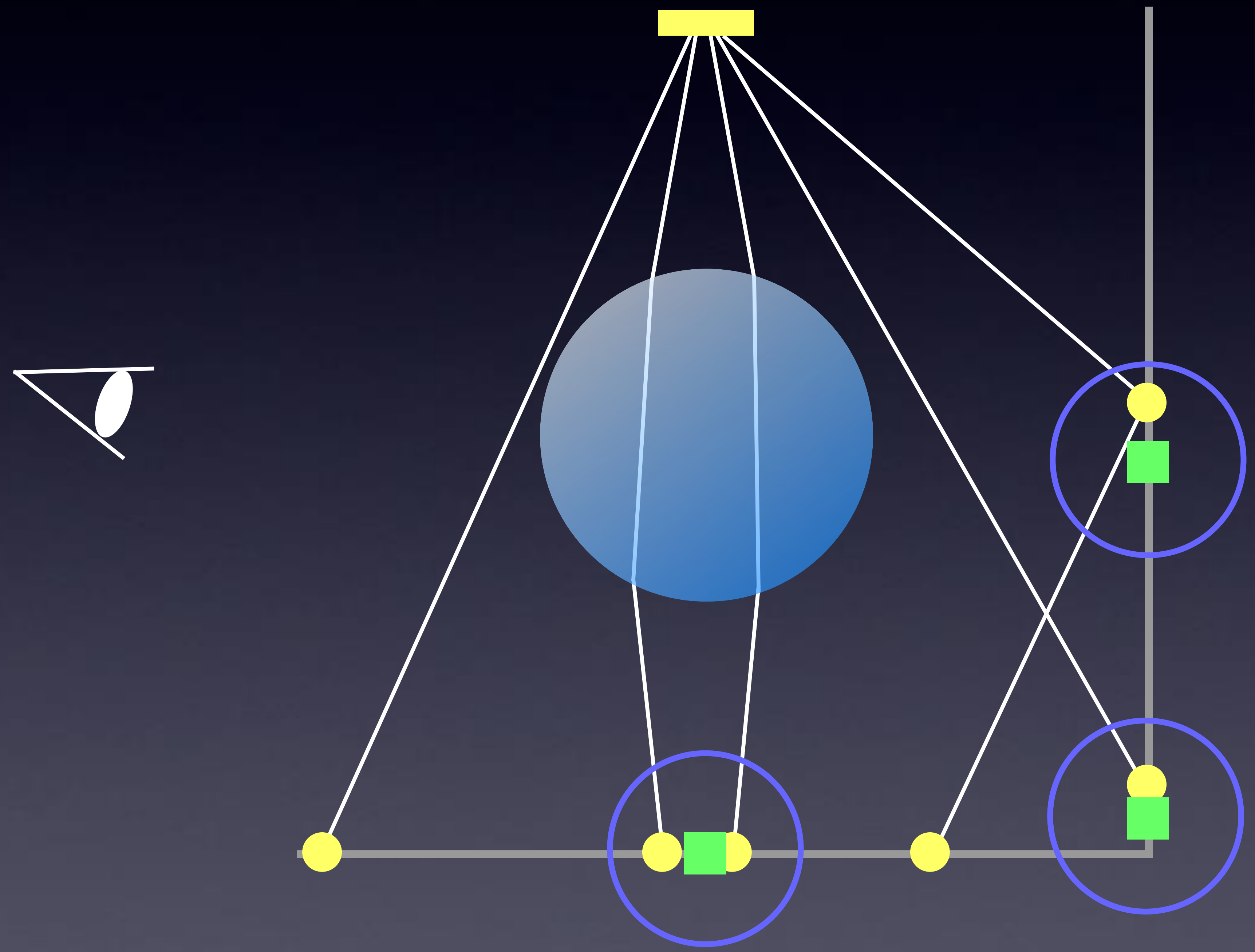
Progressive Photon Mapping - Initial Pass



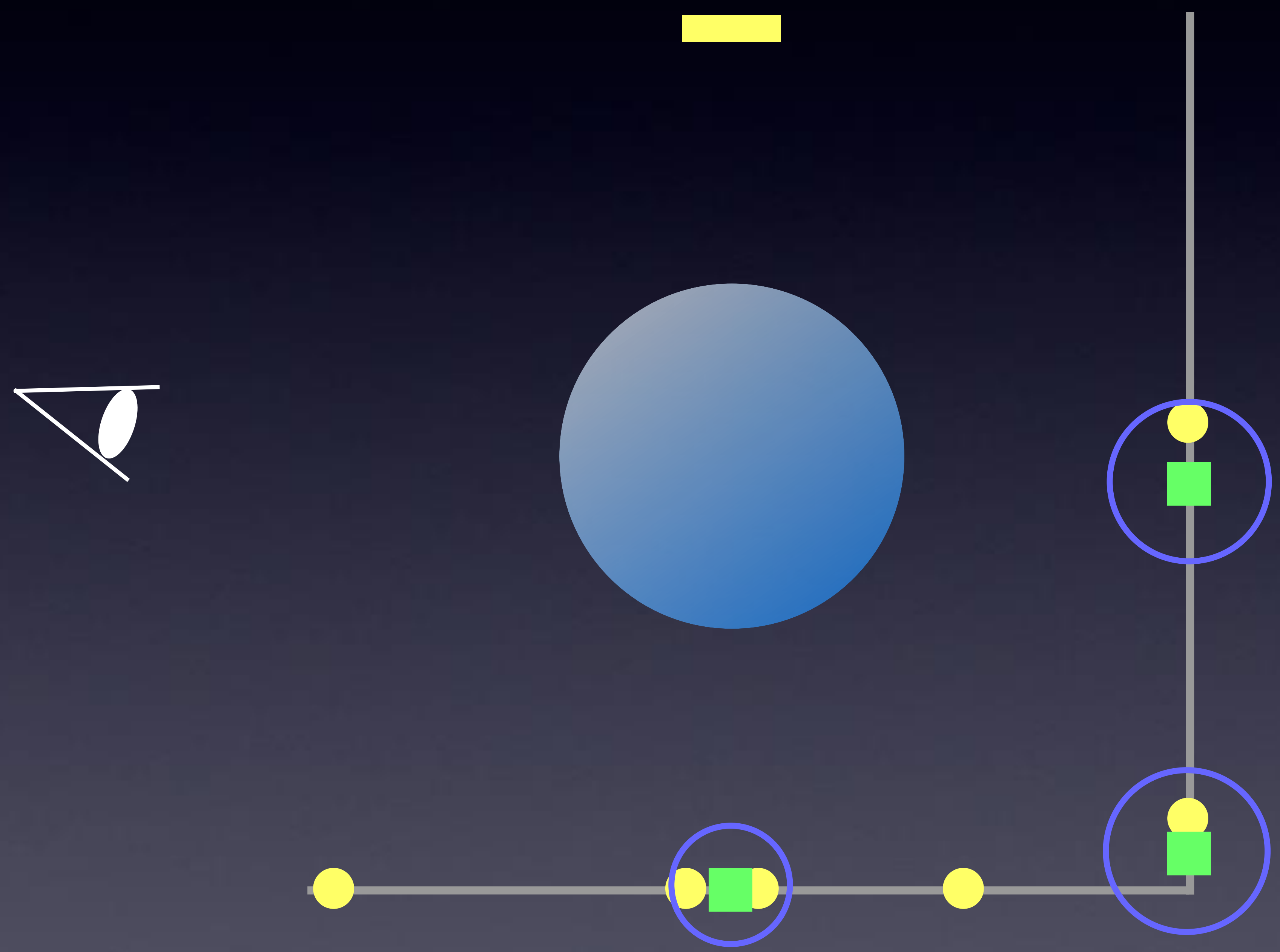
Progressive Photon Mapping - Initial Pass



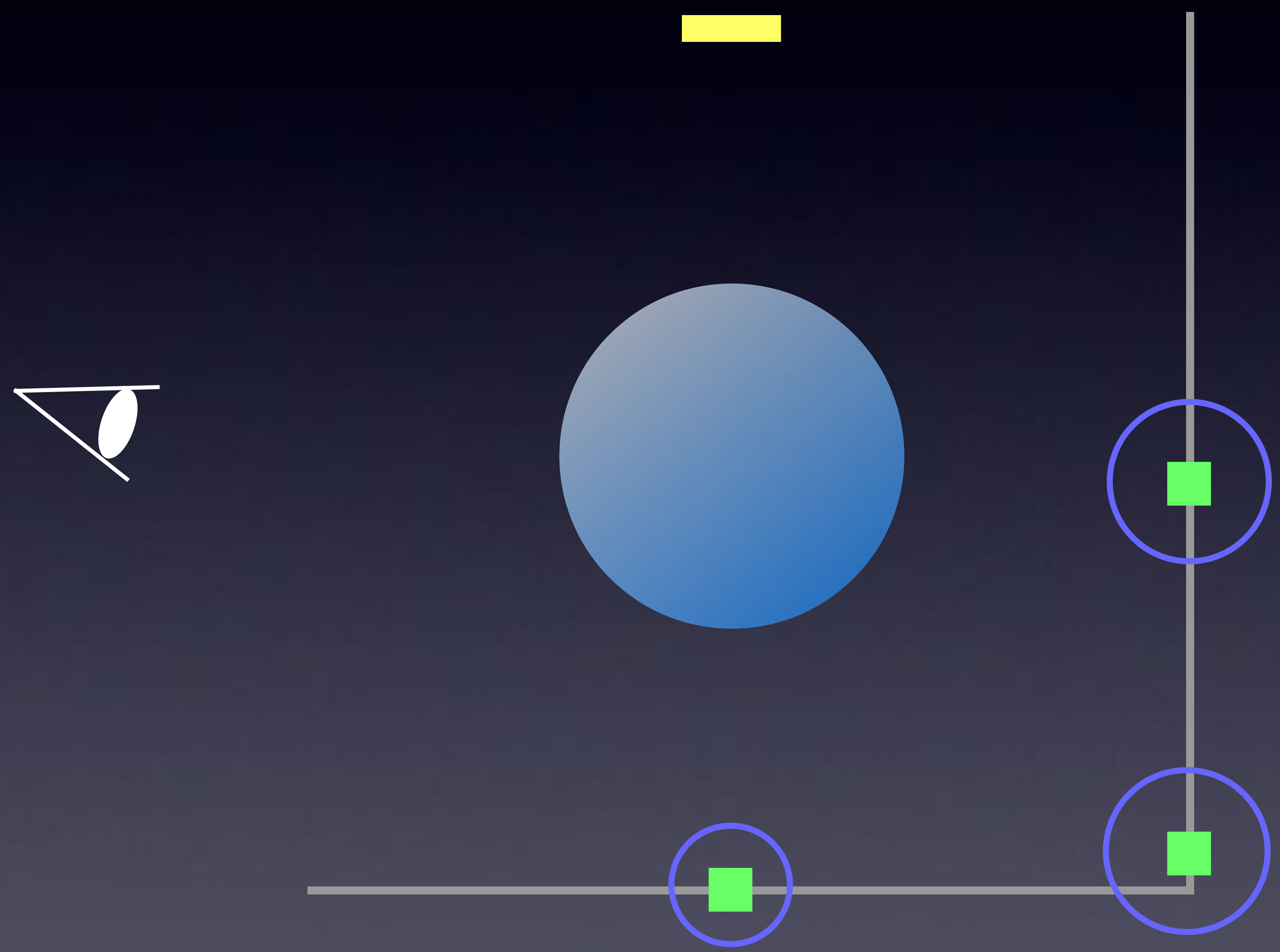
Progressive Photon Mapping - 1st Refinement Pass



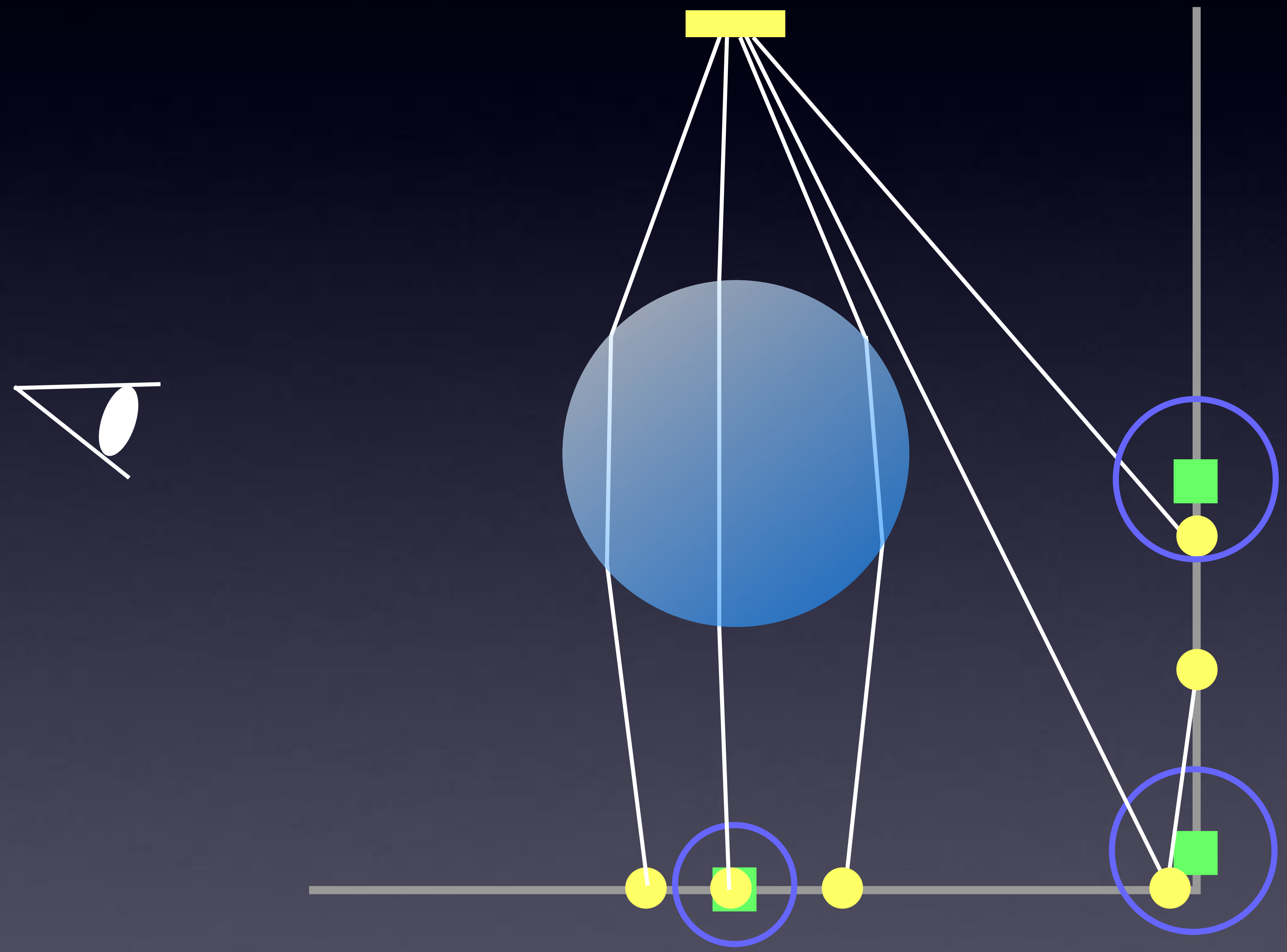
Progressive Photon Mapping - 1st Refinement Pass



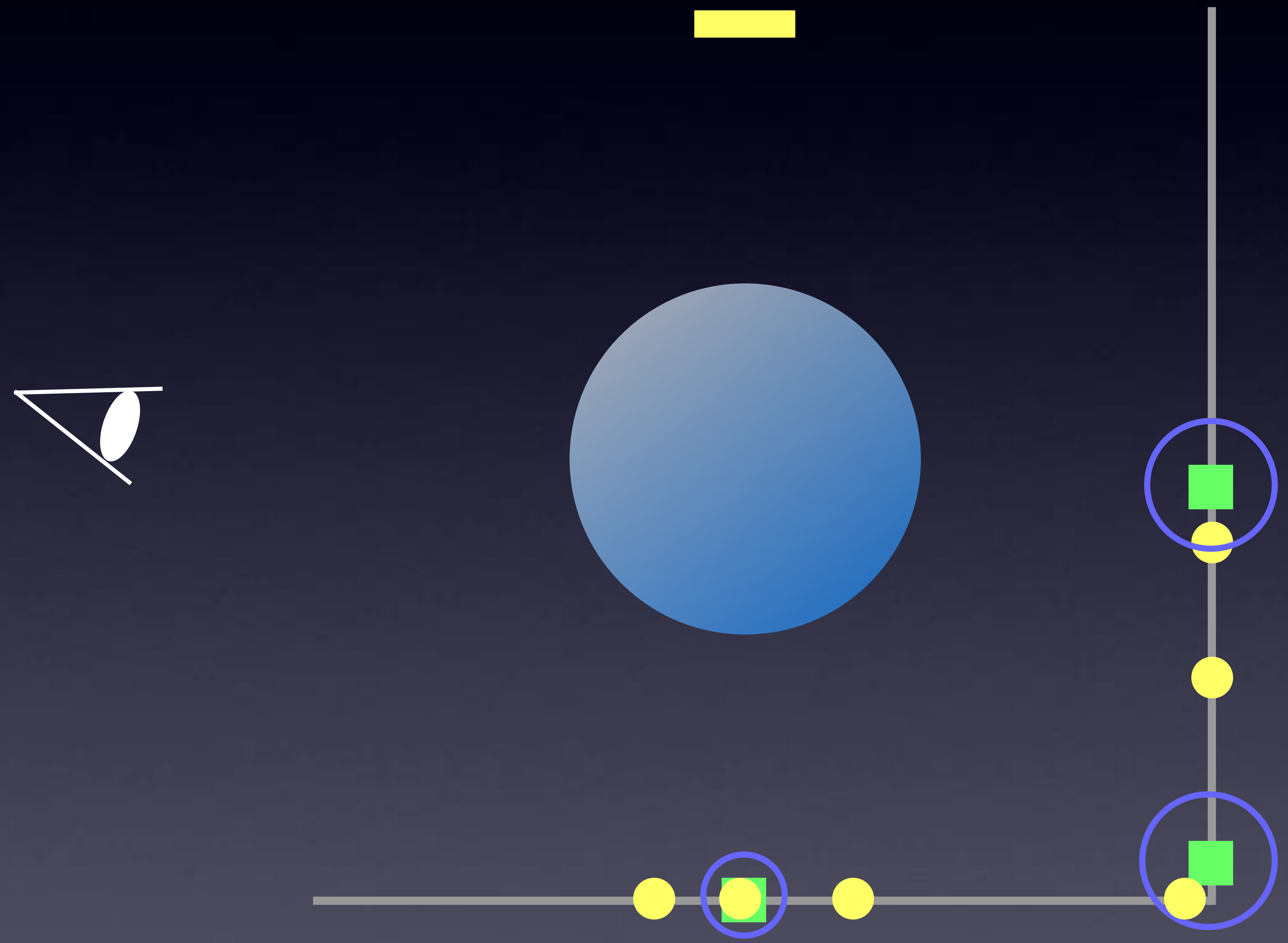
Progressive Photon Mapping - 1st Refinement Pass



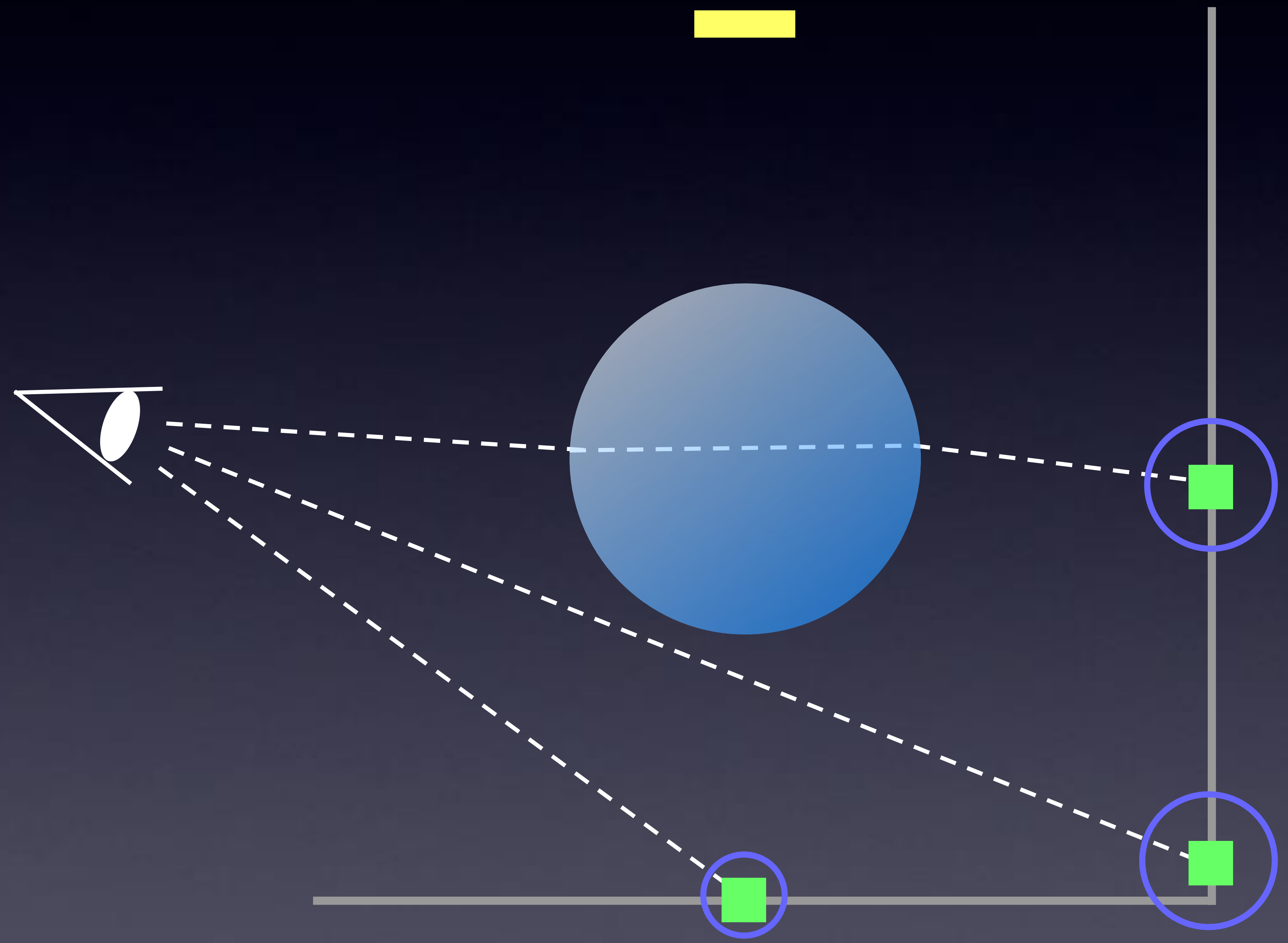
Progressive Photon Mapping - 2nd Refinement Pass



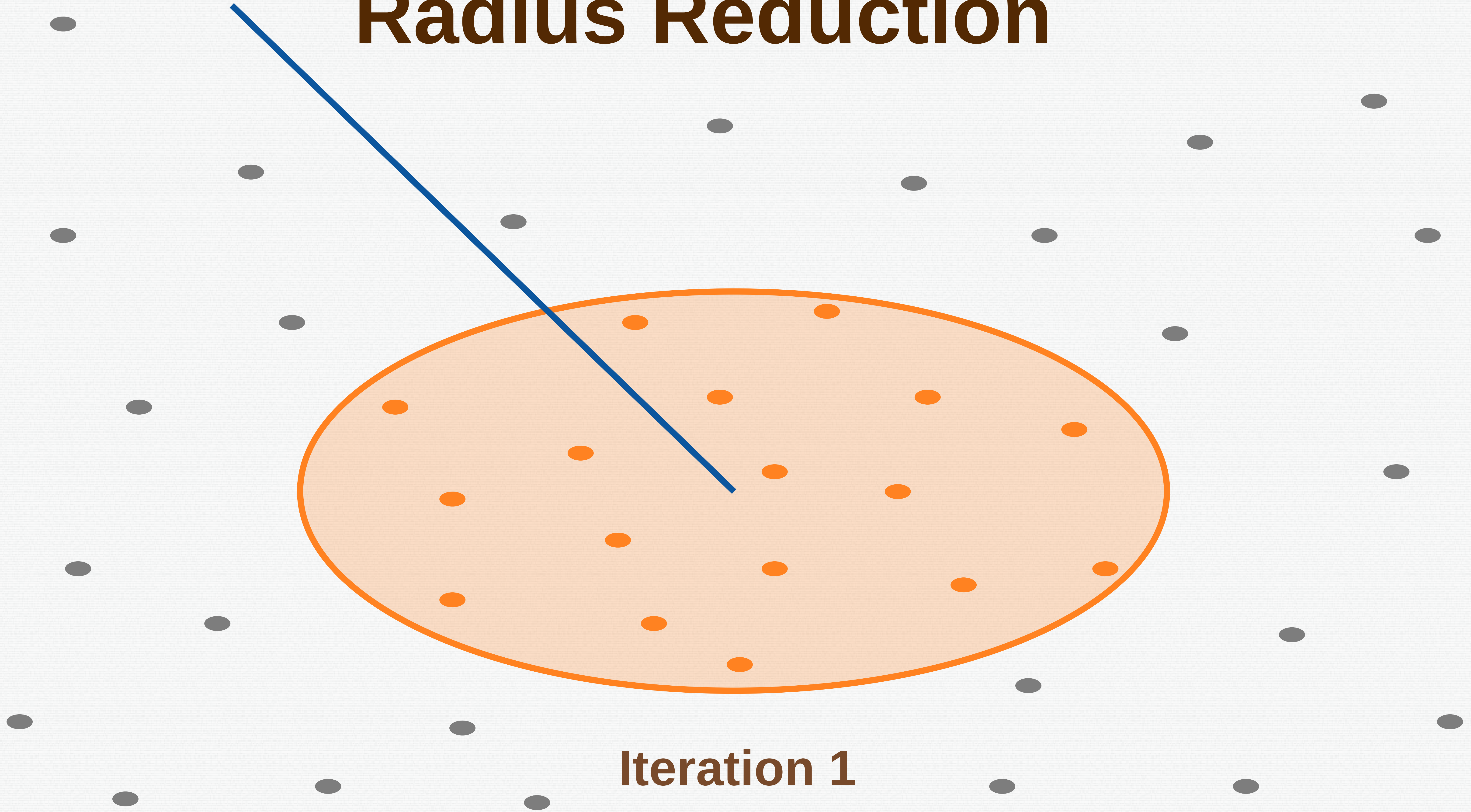
Progressive Photon Mapping - 2nd Refinement Pass



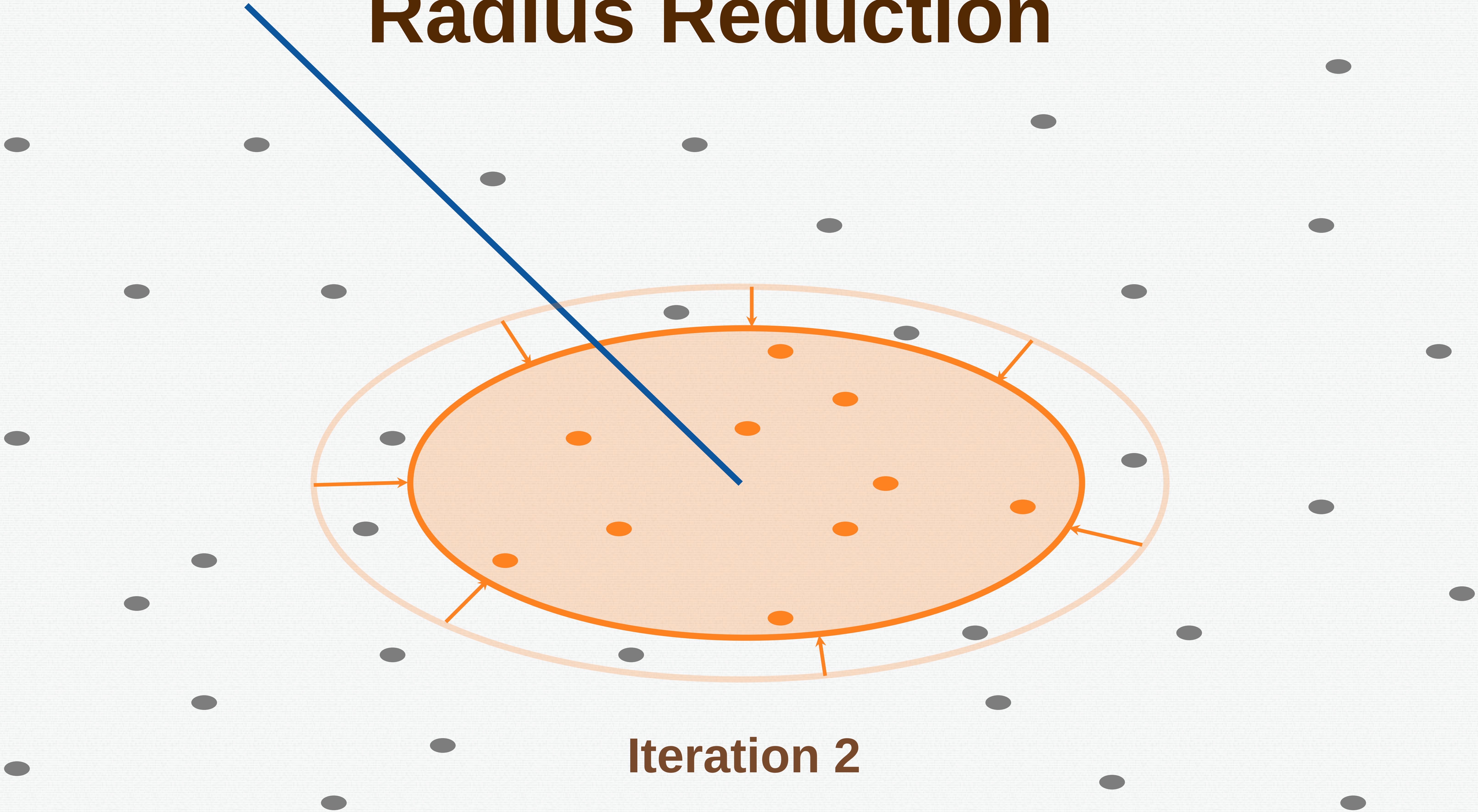
Progressive Photon Mapping - Rendering



Radius Reduction

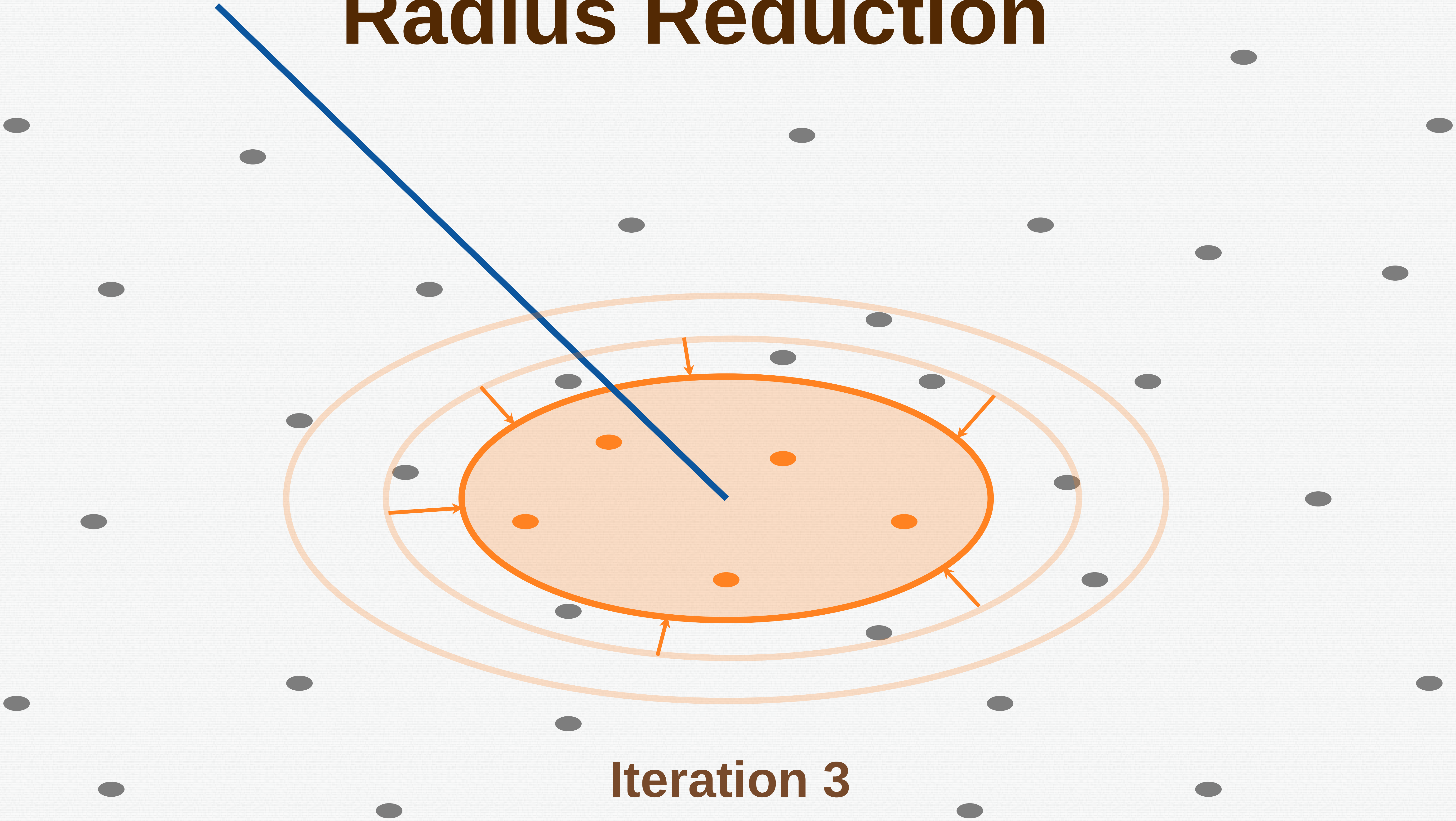


Radius Reduction



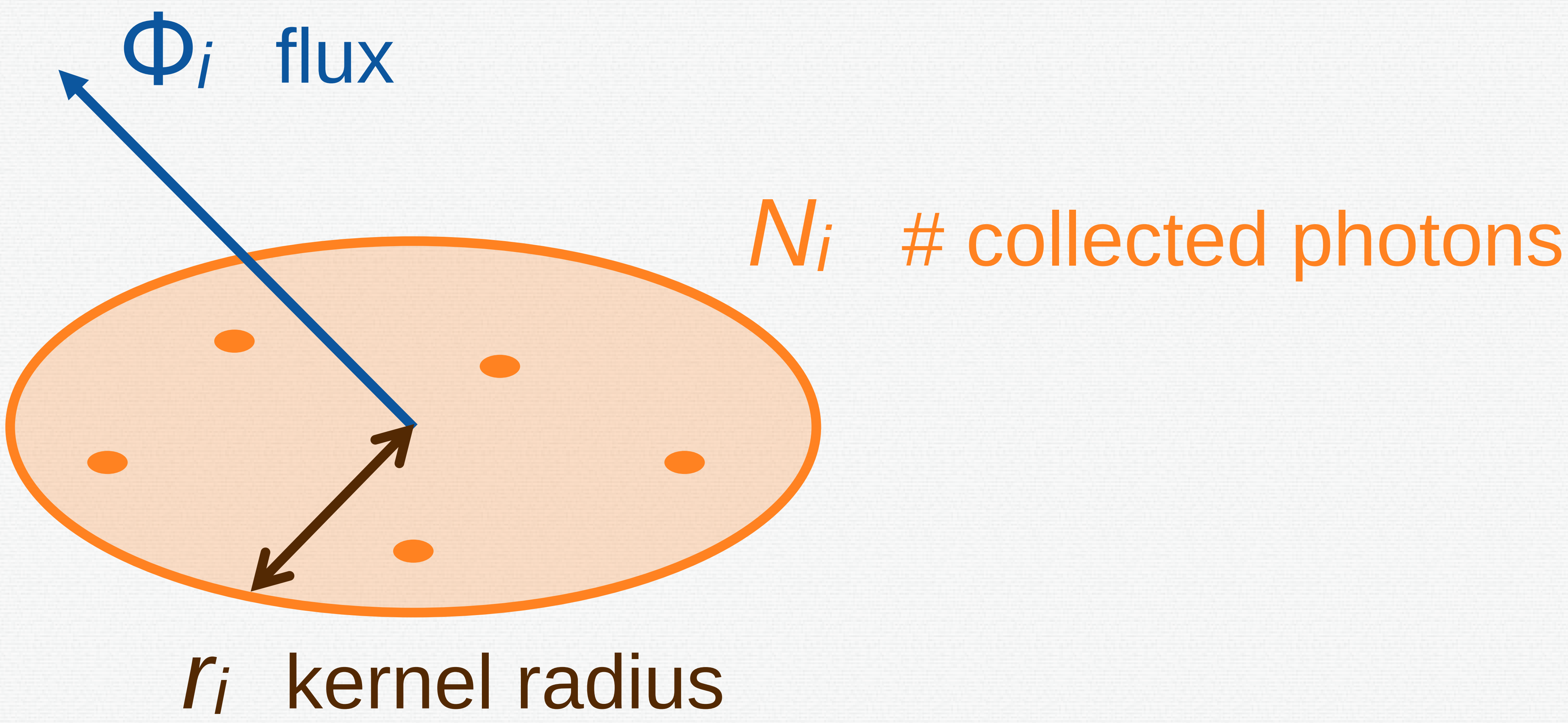
Iteration 2

Radius Reduction



Iteration 3

Locations with Statistics



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

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.

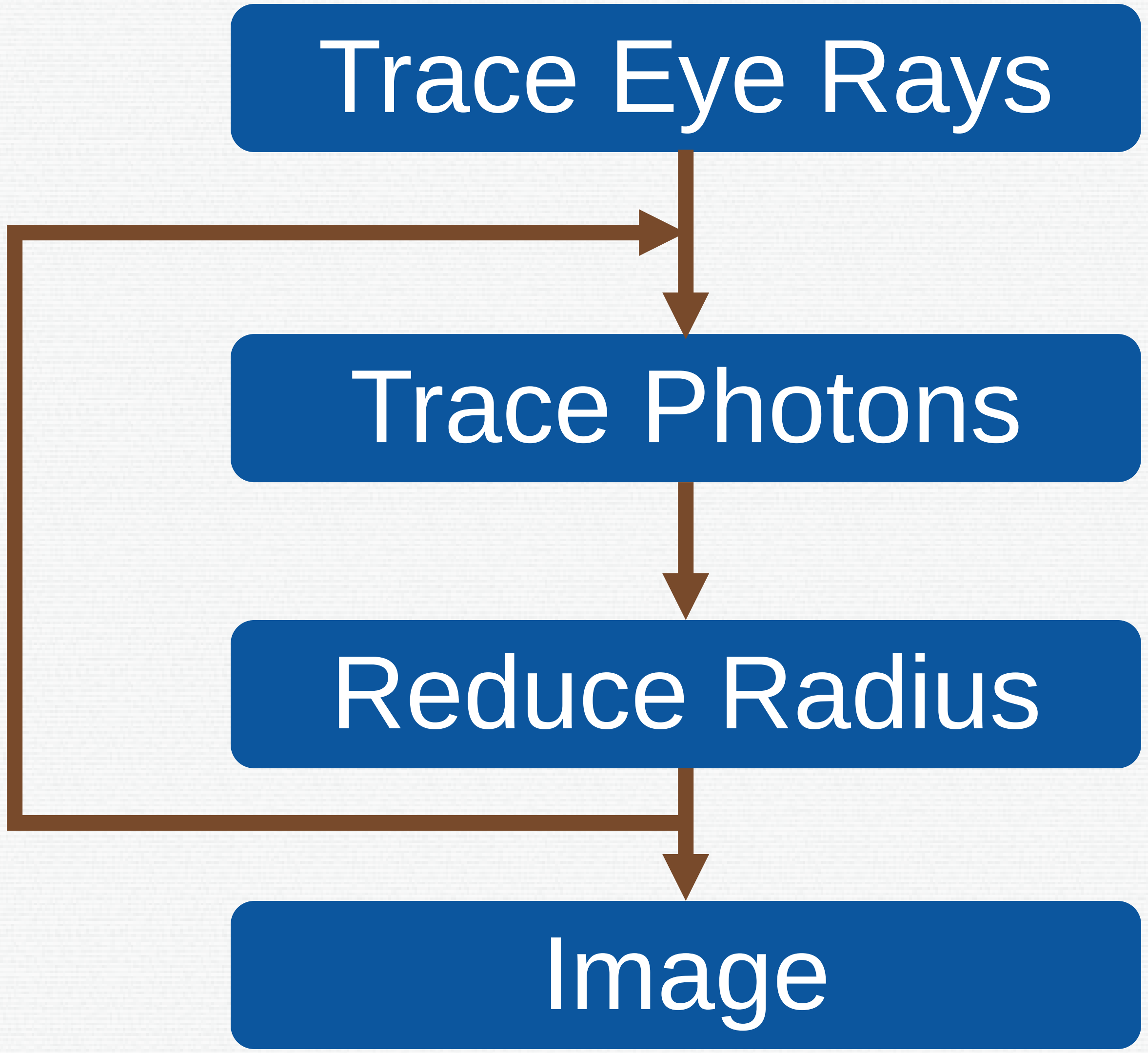
Hachisuka et al. [2008] observed out that specular-diffuse-specular paths (SDS paths in the light path notation) are particularly problematic for the existing unbiased methods. An example of an SDS path is a light path due to the combination of specular materials and a light bulb. Photon mapping [Jensen 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.

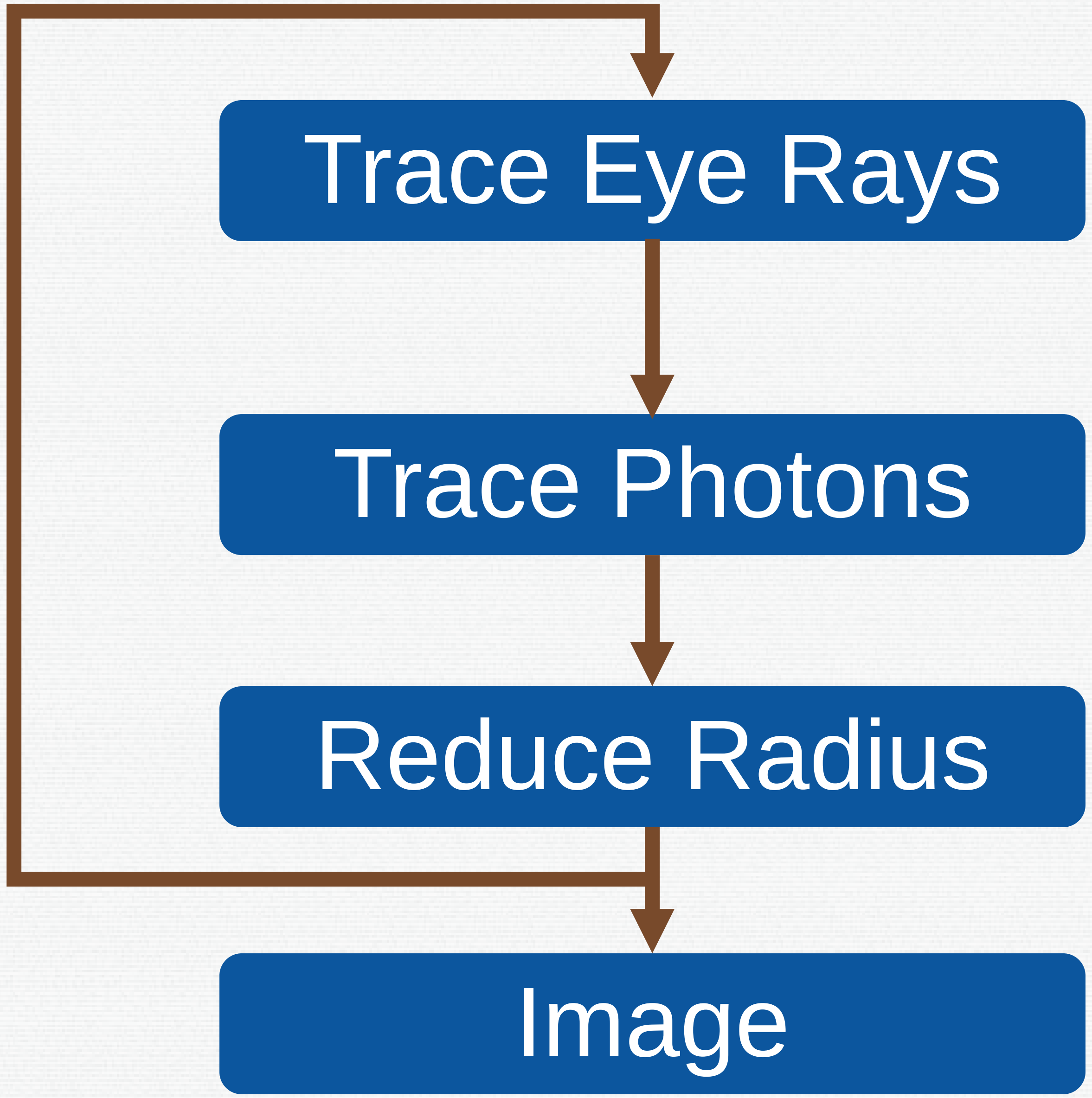
In this paper, we present a new formulation of progressive photon mapping that enables computing the correct average radiance value over a region. Our formulation requires a simple algorithmic modification, which consists of adding a distributed ray tracing pass after each photon pass in progressive 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

Stochastic PPM



Stochastic PPM



SIGGRAPH2011

Progressive Photon Mapping: A Probabilistic Approach

Claude Knaus and Matthias Zwicker

University of Bern

cgg
computer graphics group

Our Probabilistic Approach

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

Radiance Estimation

Stored Photons

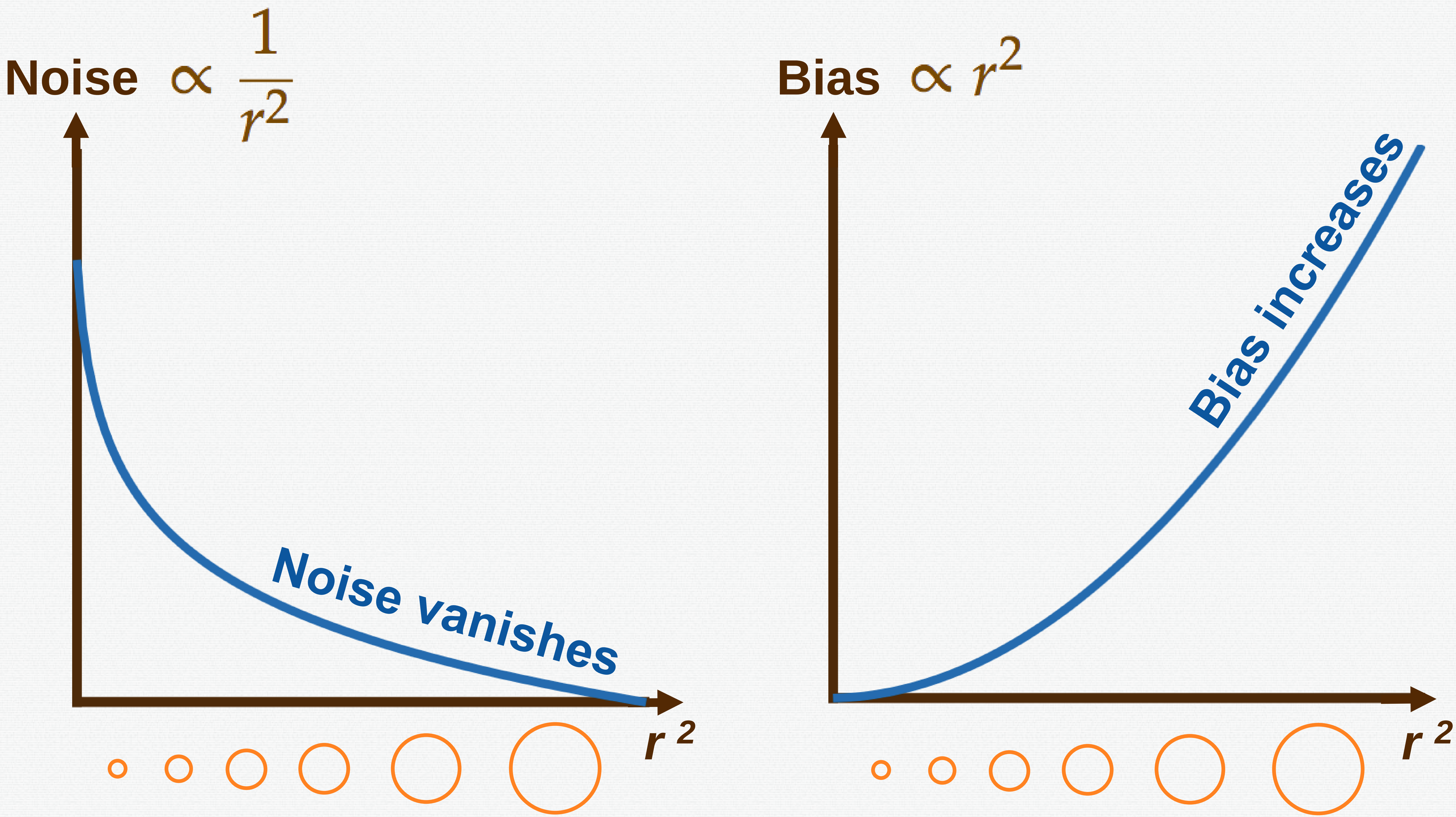
$$L(x) \approx \frac{1}{N_e} \sum_{i=1}^{N_s} k_r(x_i - x)\gamma_i$$

Emitted Photons

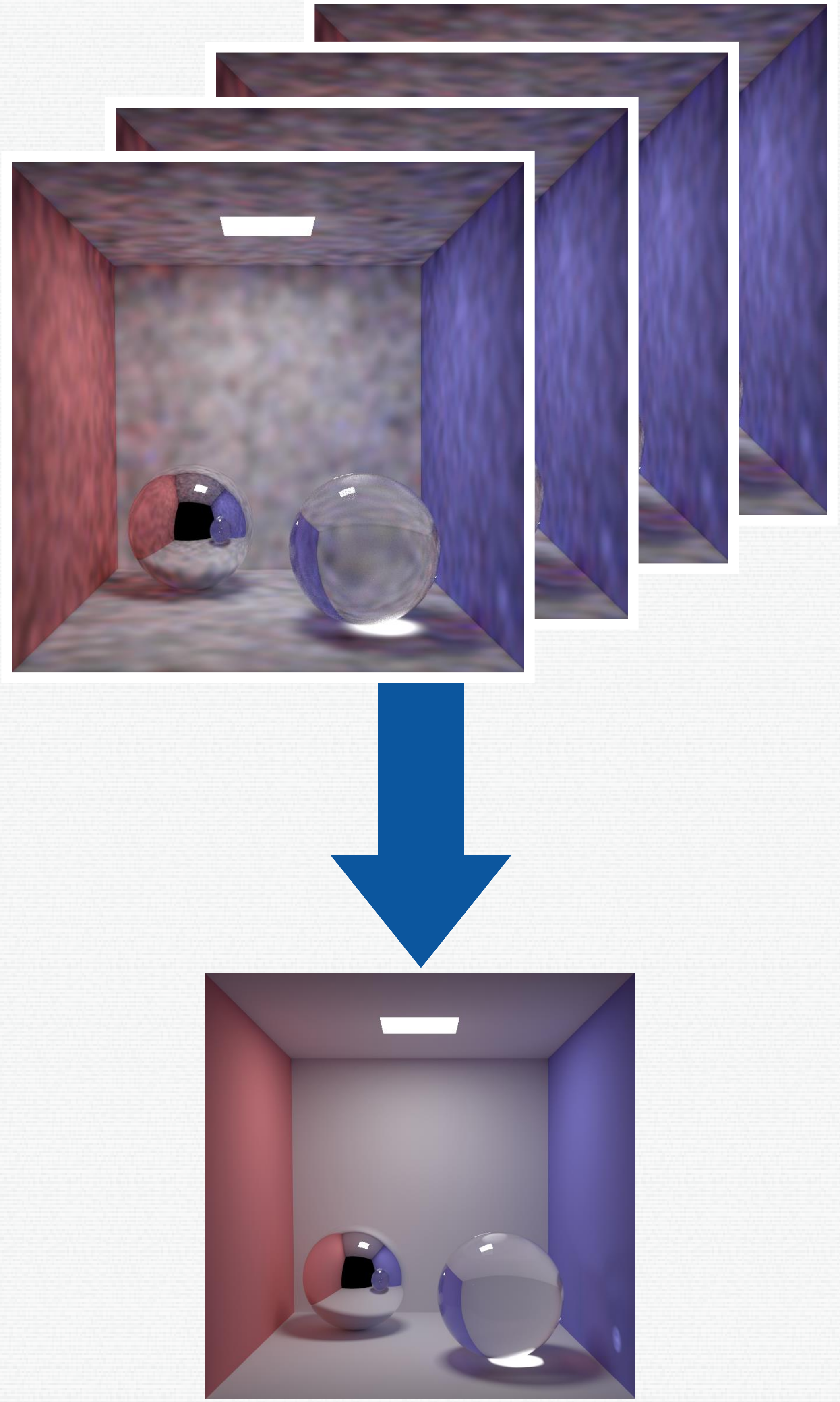
Kernel with Radius *r*

Photon Position

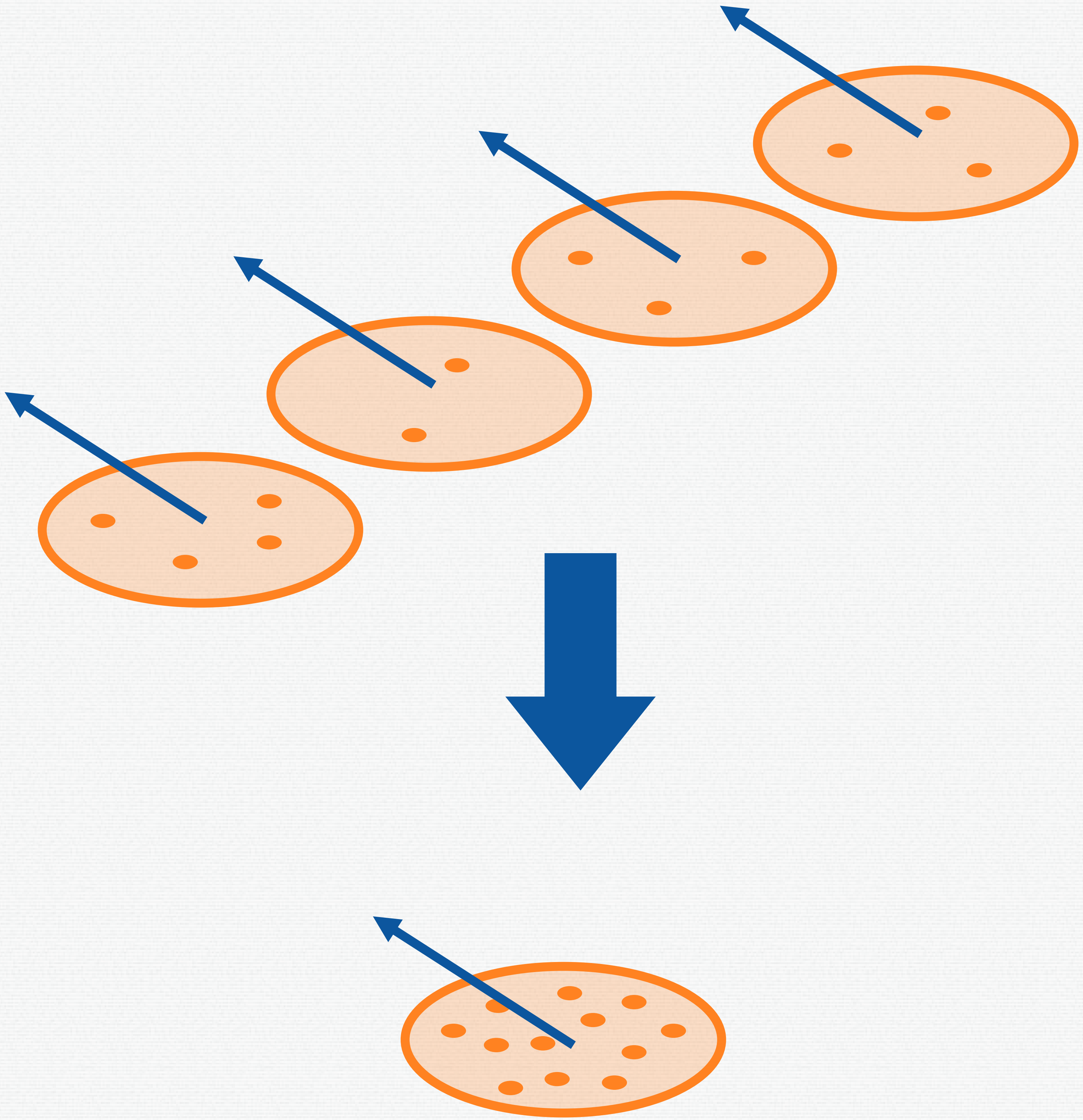
Photon Power



Noise $\propto \frac{1}{r^2}$
Noise vanishes
r^2
Bias $\propto r^2$
Bias increases
r^2

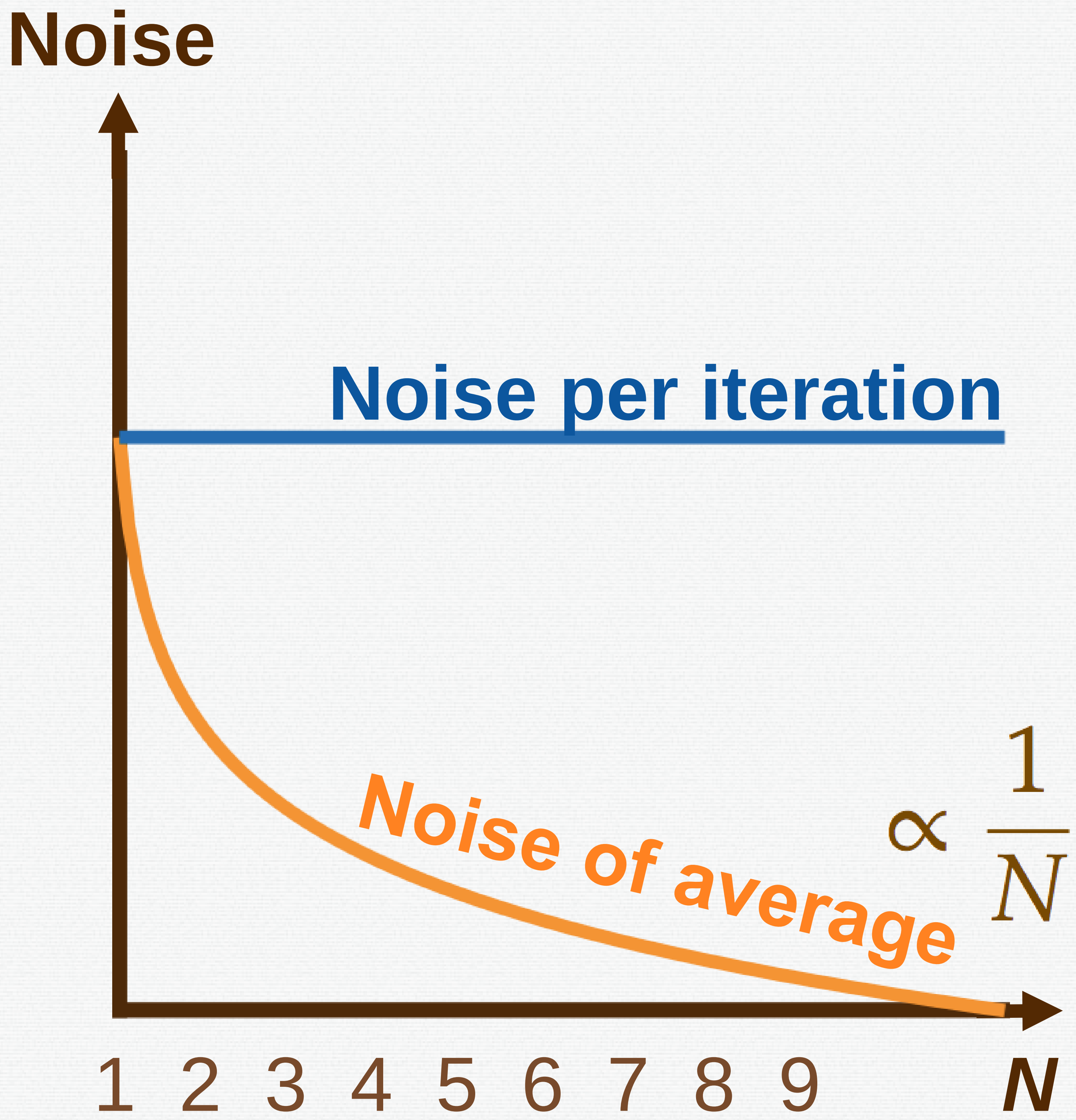


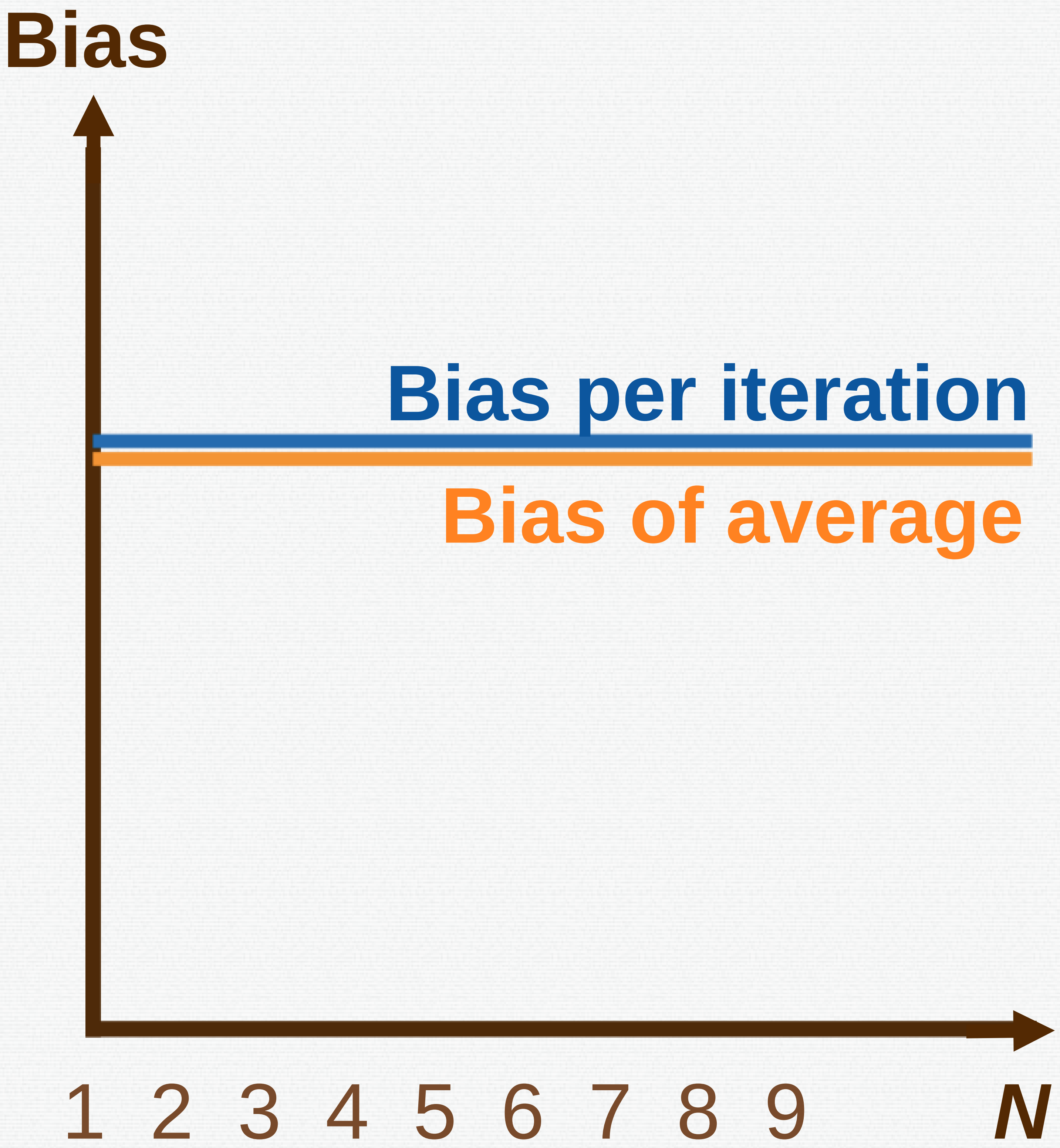
Averaged Image



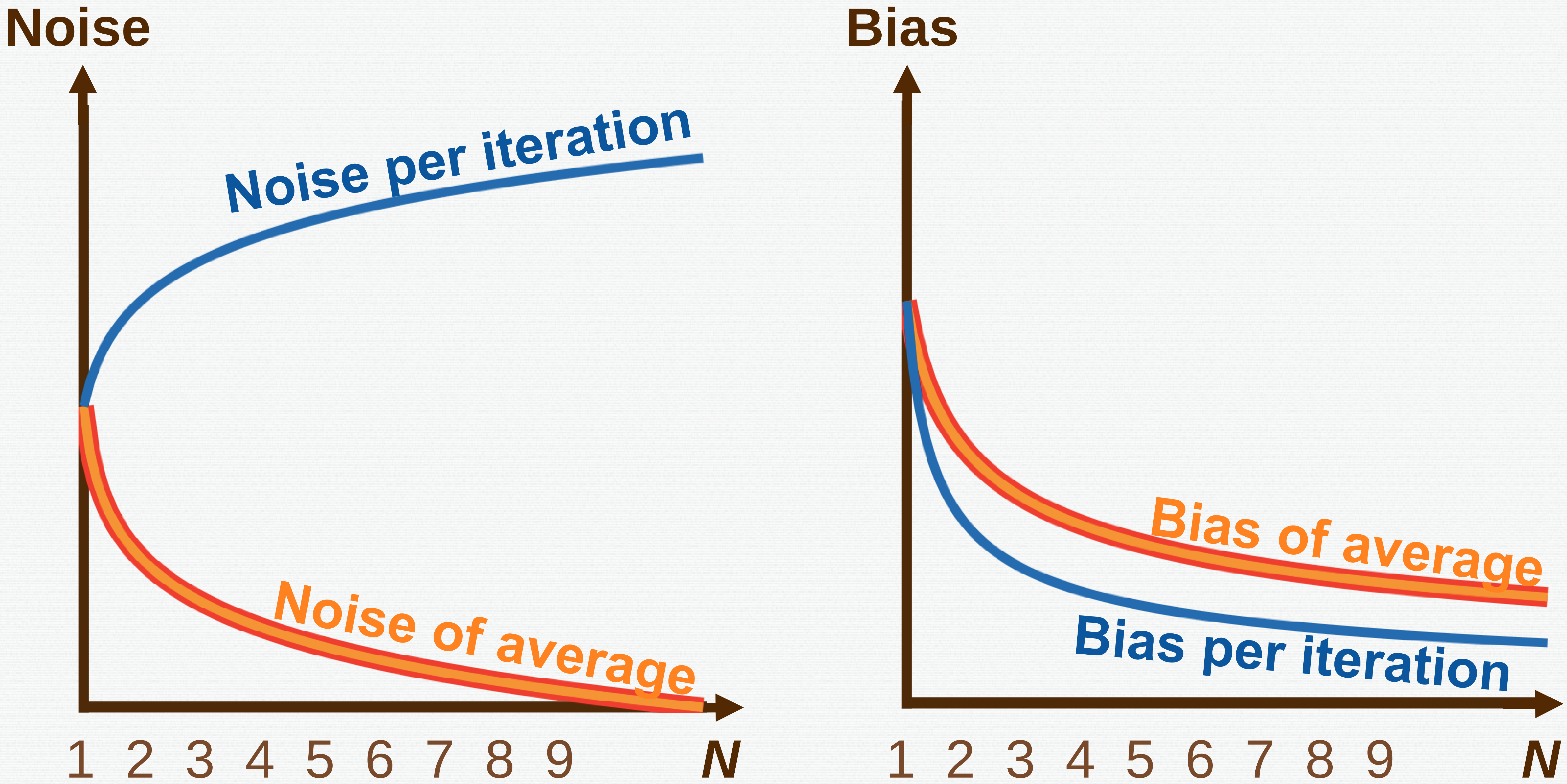
Averaged Radiance Estimate

Averaged Radiance Estimates



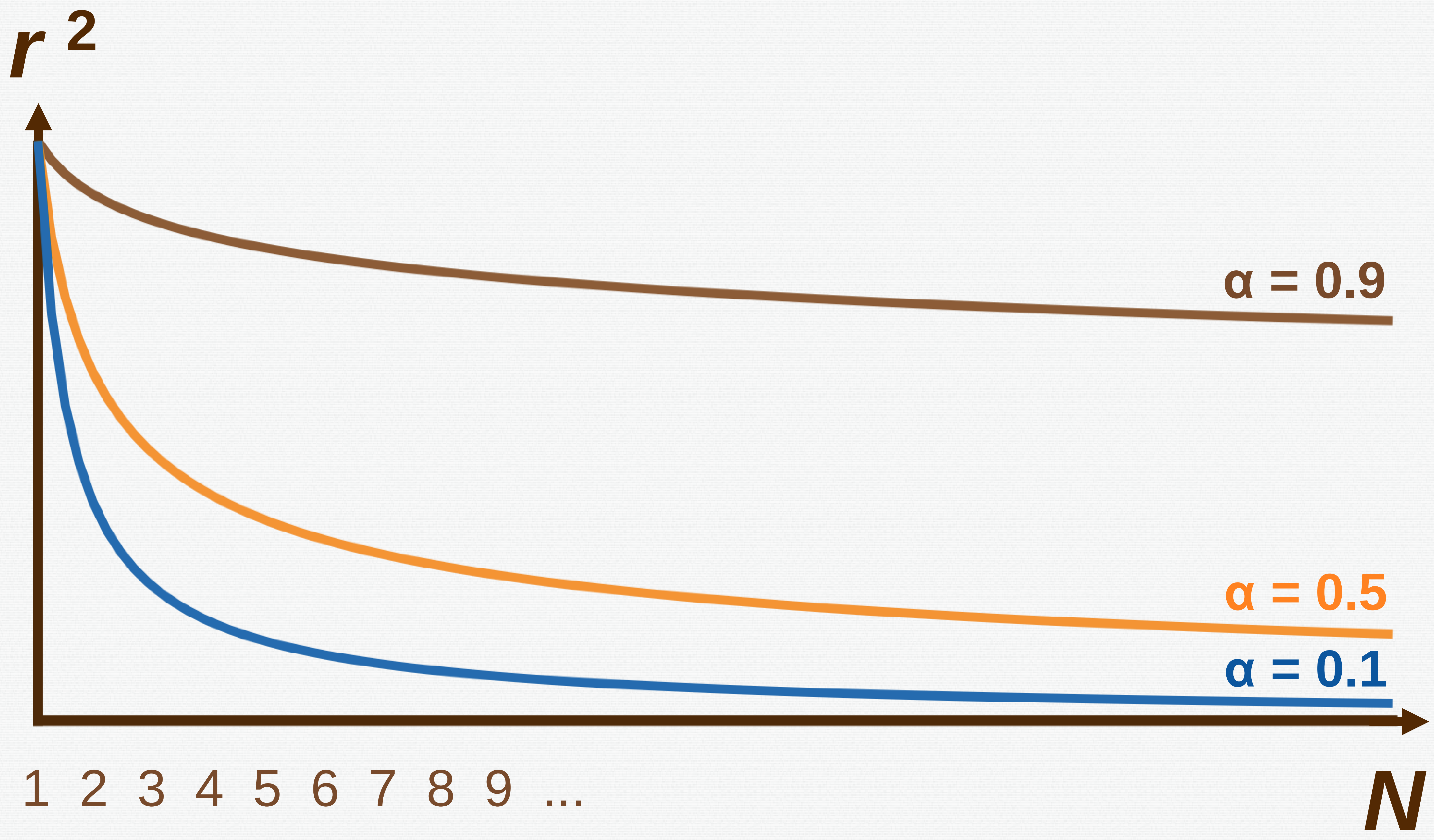


Averaging + Radius Reduction



Radius Sequence

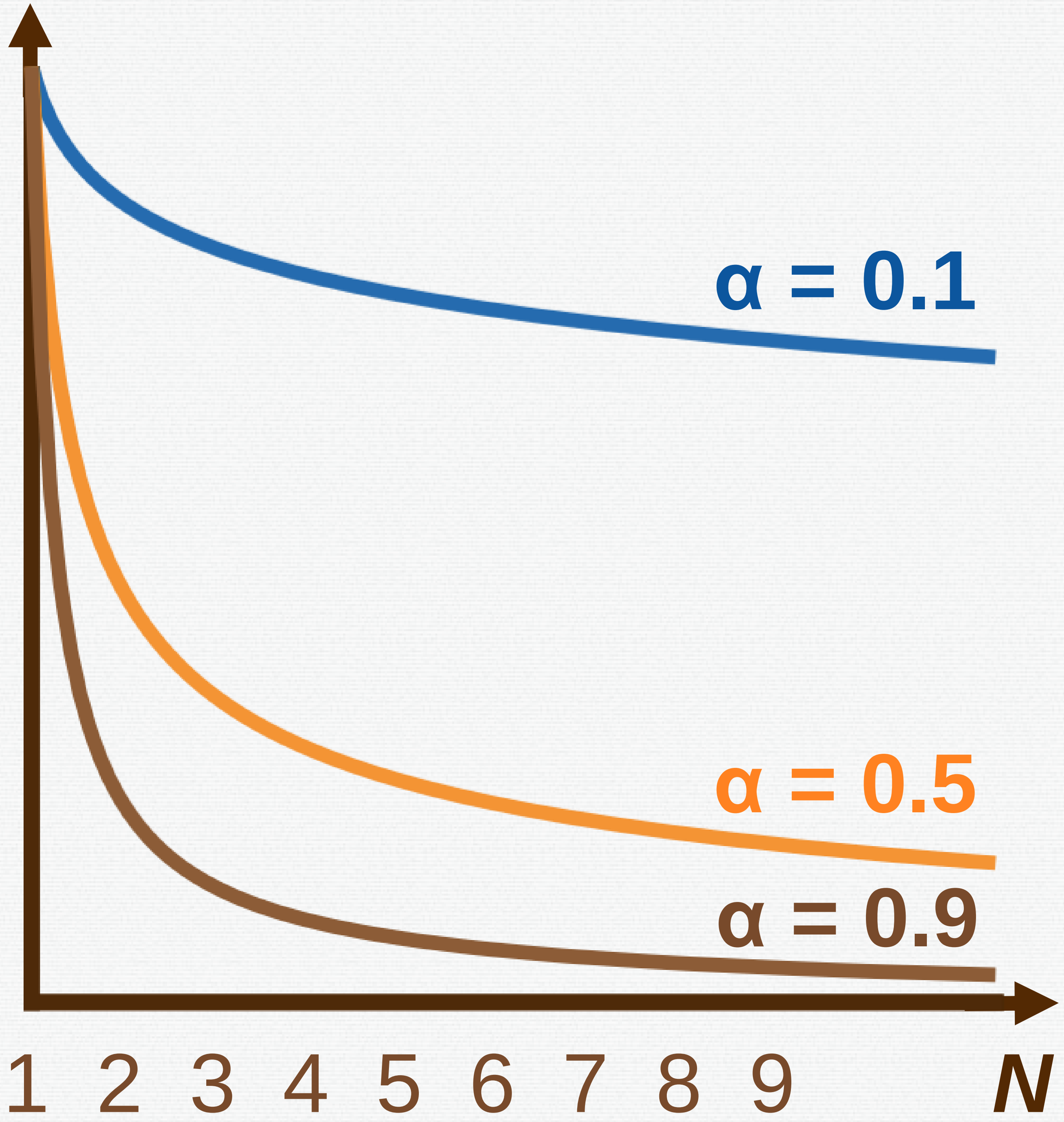
$$\frac{r_{i+1}^2}{r_i^2} = \frac{i+\alpha}{i+1}$$



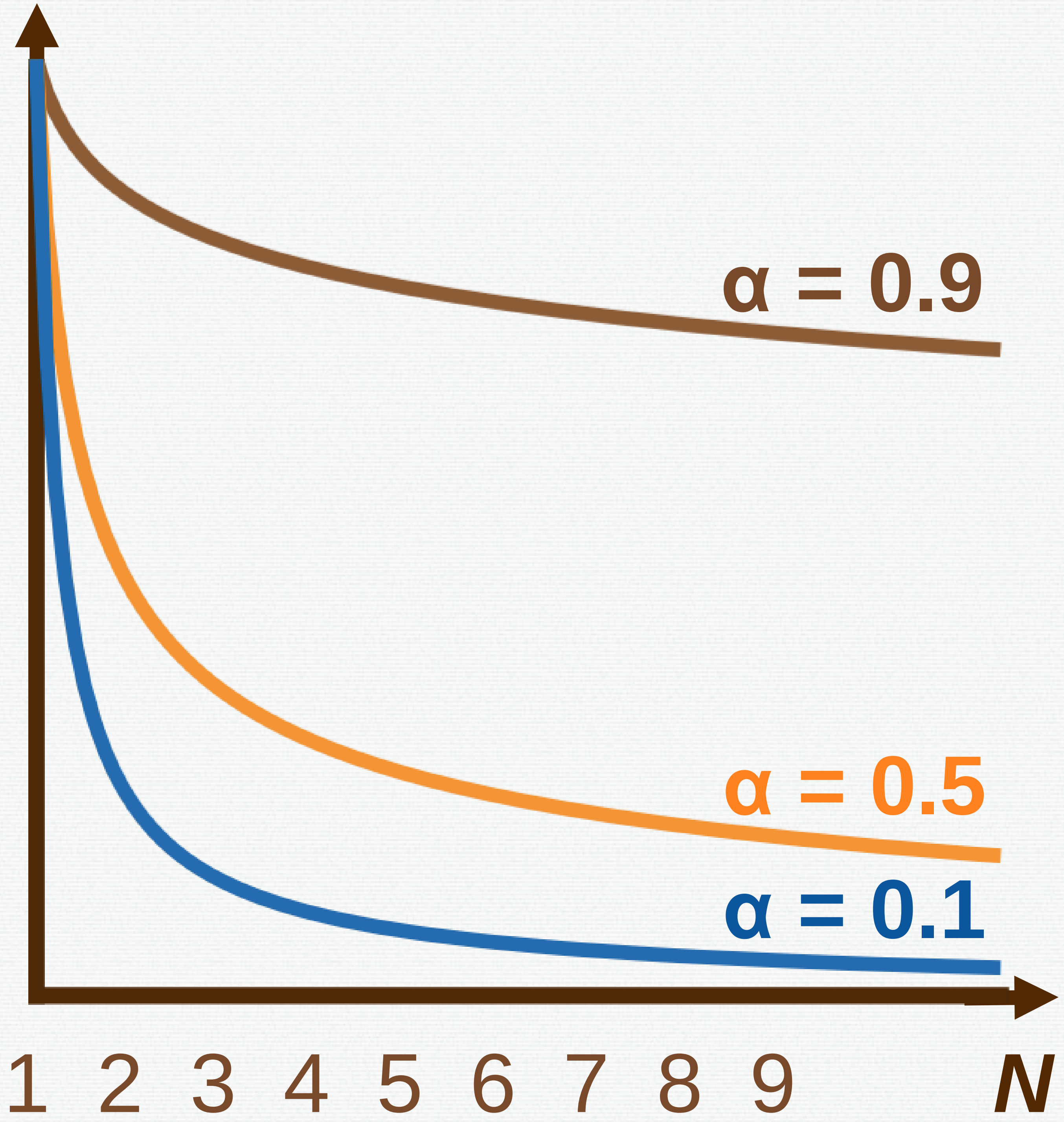
r 2
α = 0.9
α = 0.5
α = 0.1
1 2 3 4 5 6 7 8 9 ...
N

Asymptotic Convergence

Noise of average $\propto 1/N^{\alpha}$

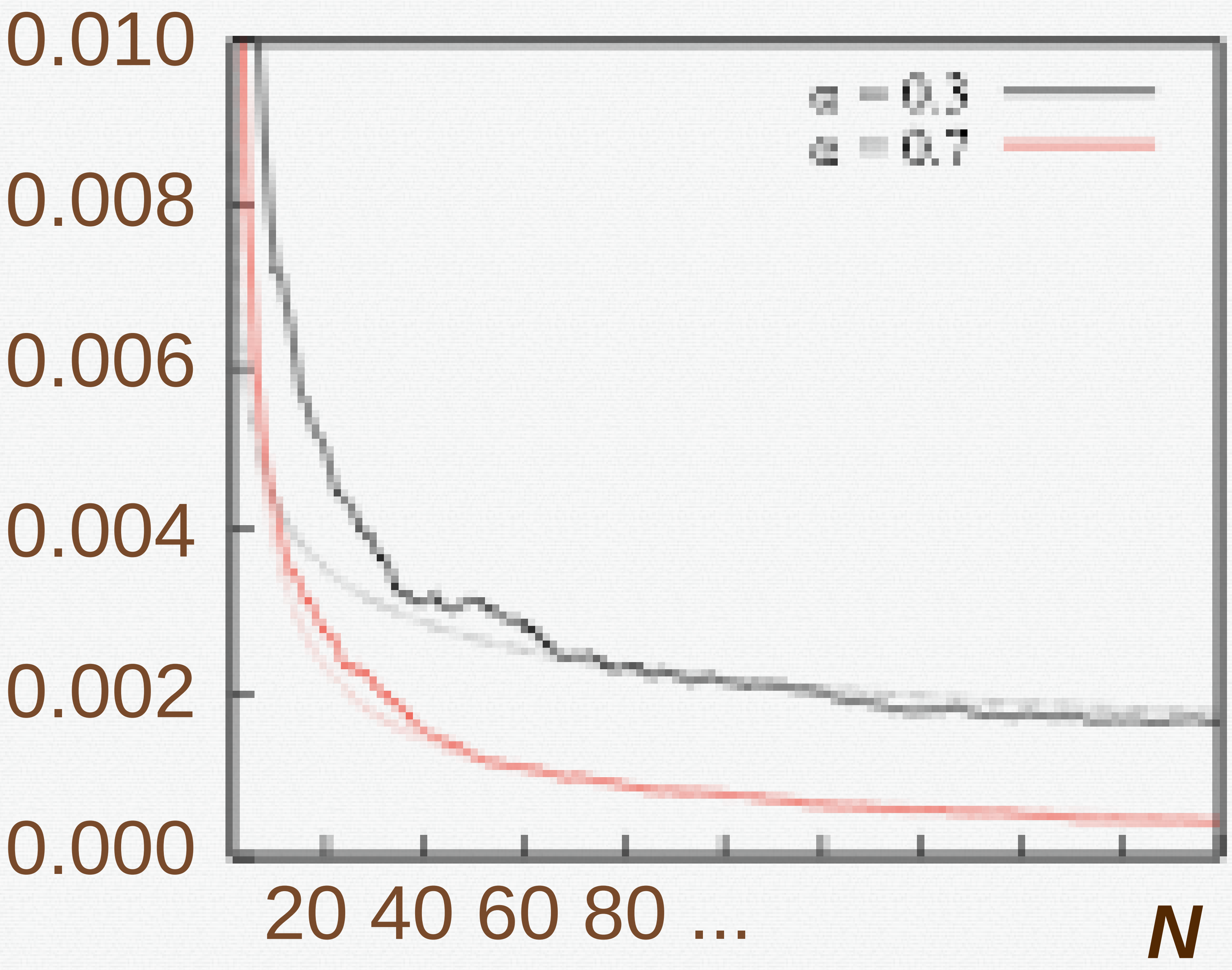


Bias of average $\propto 1/N^{1-\alpha}$

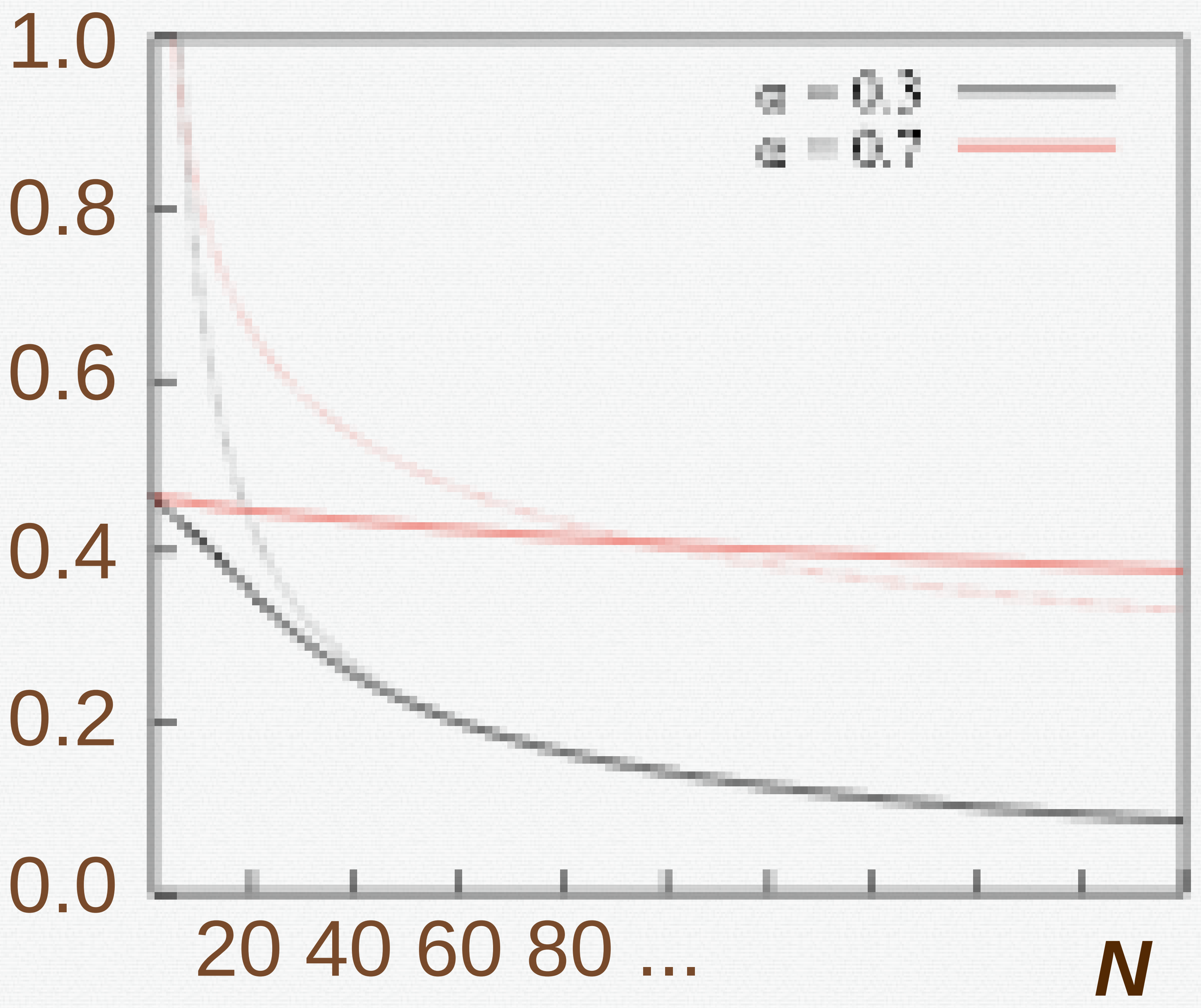


Empirical Validation

Noise of average



Bias of average



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!

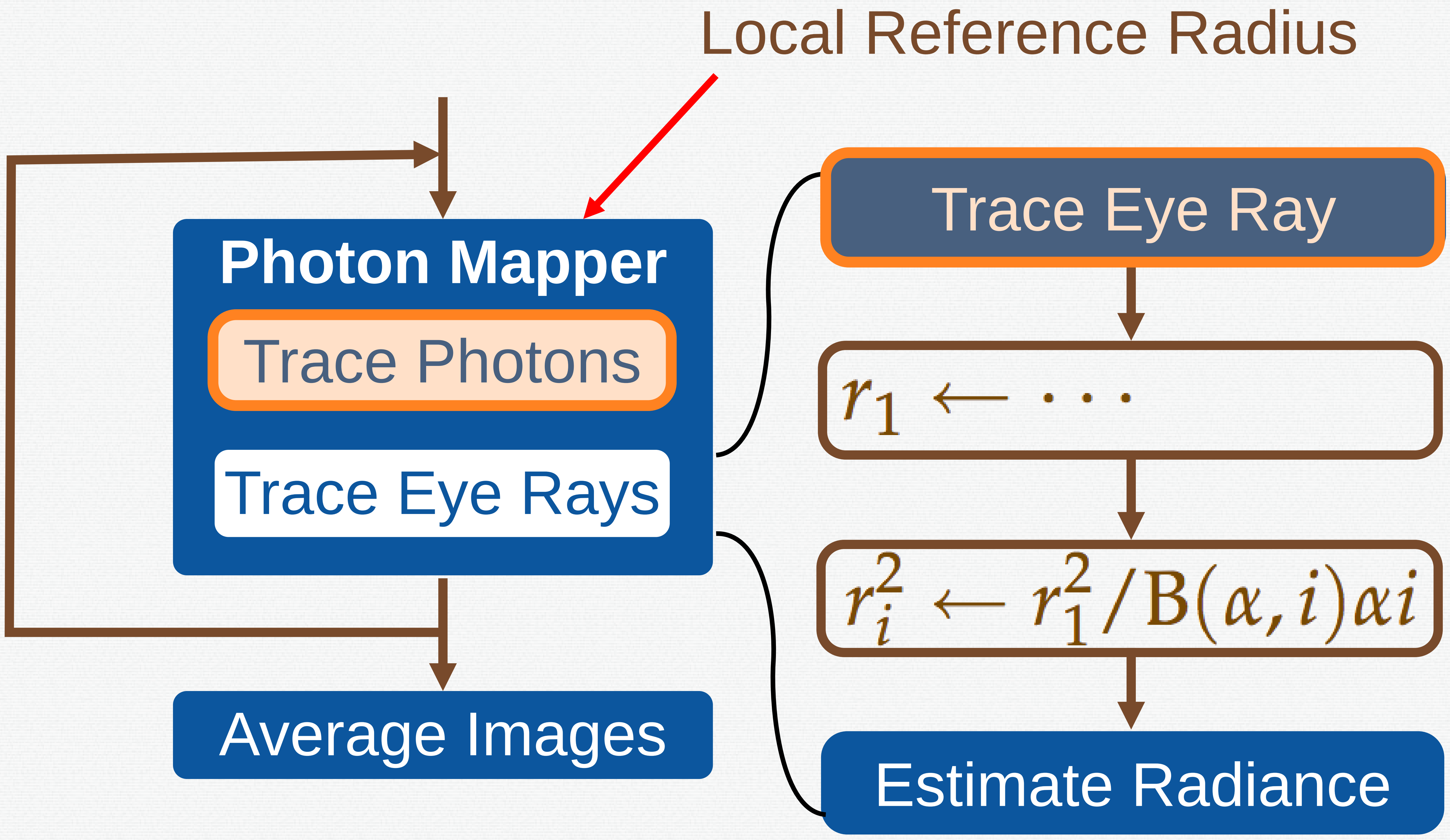
Radius Sequence (Explicit)

Reference Radius

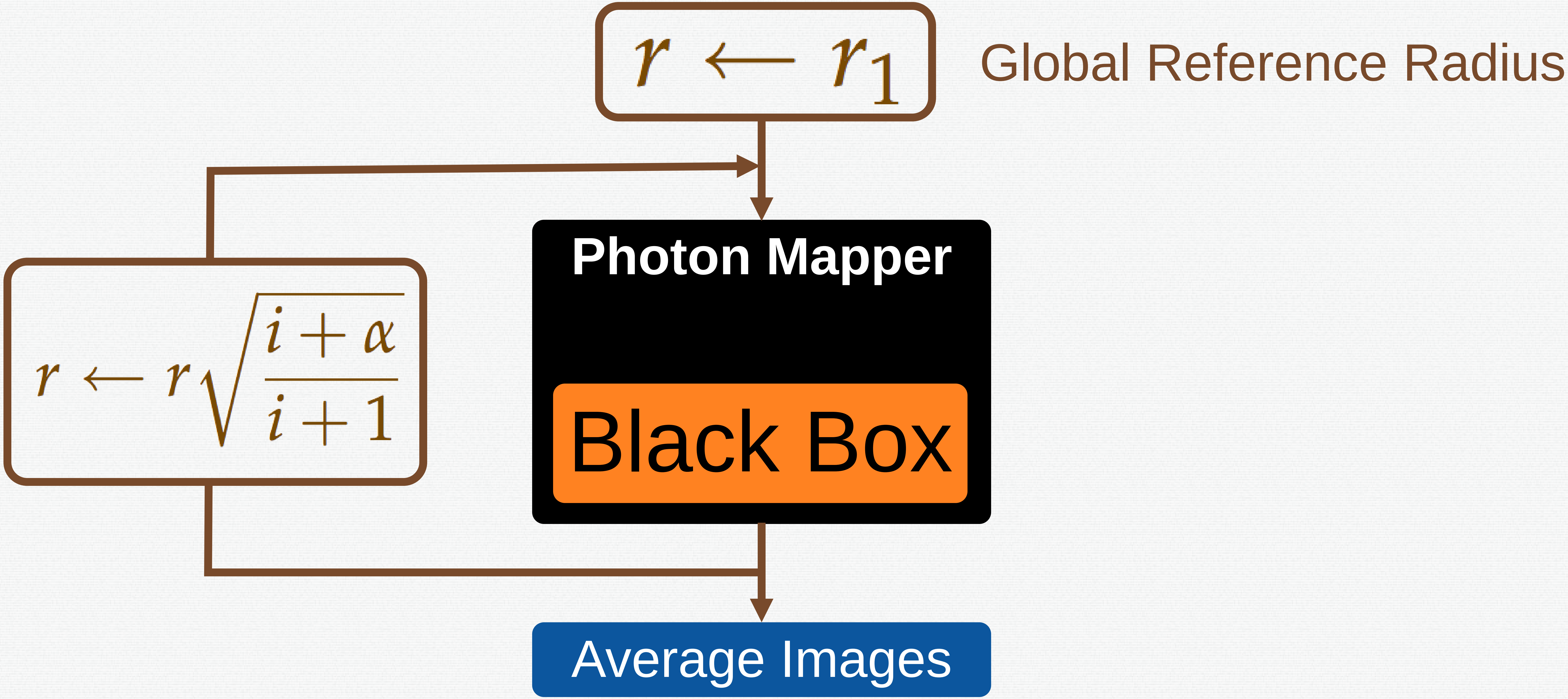
$$r_i^2 = \frac{r_1^2}{\mathrm{B}(\alpha, i)\alpha i}$$

Beta Function

Our Algorithm



Script


$r \leftarrow r_1$
Global Reference Radius
$r \leftarrow r\sqrt{\frac{i+\alpha}{i+1}}$
Photon Mapper
Black Box
Average Images


1
2
3
4
5
6
1000

1

2

3

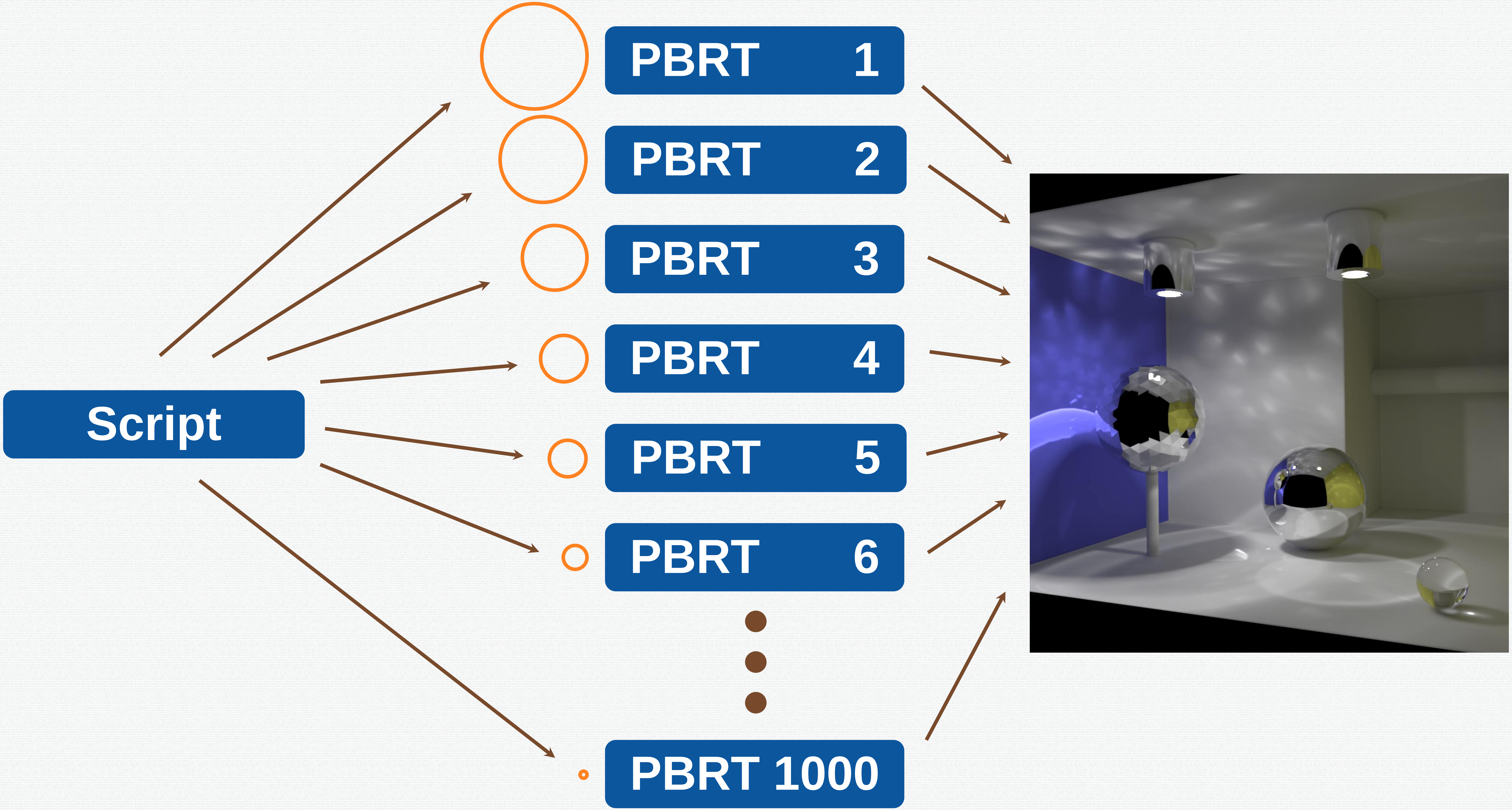
4

5

6

⋮

1000


Script
PBRT 1
PBRT 2
PBRT 3
PBRT 4
PBRT 5
PBRT 6
PBRT 1000

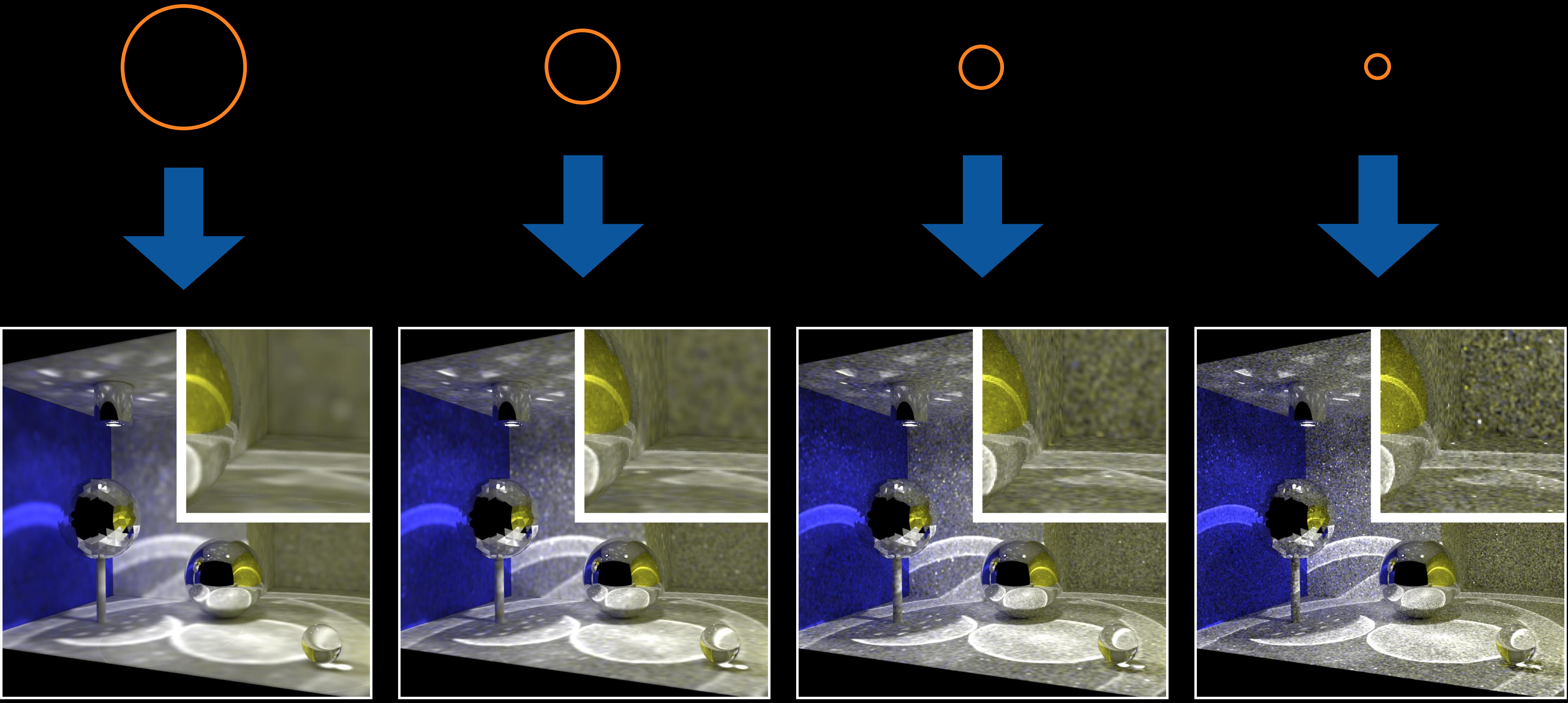
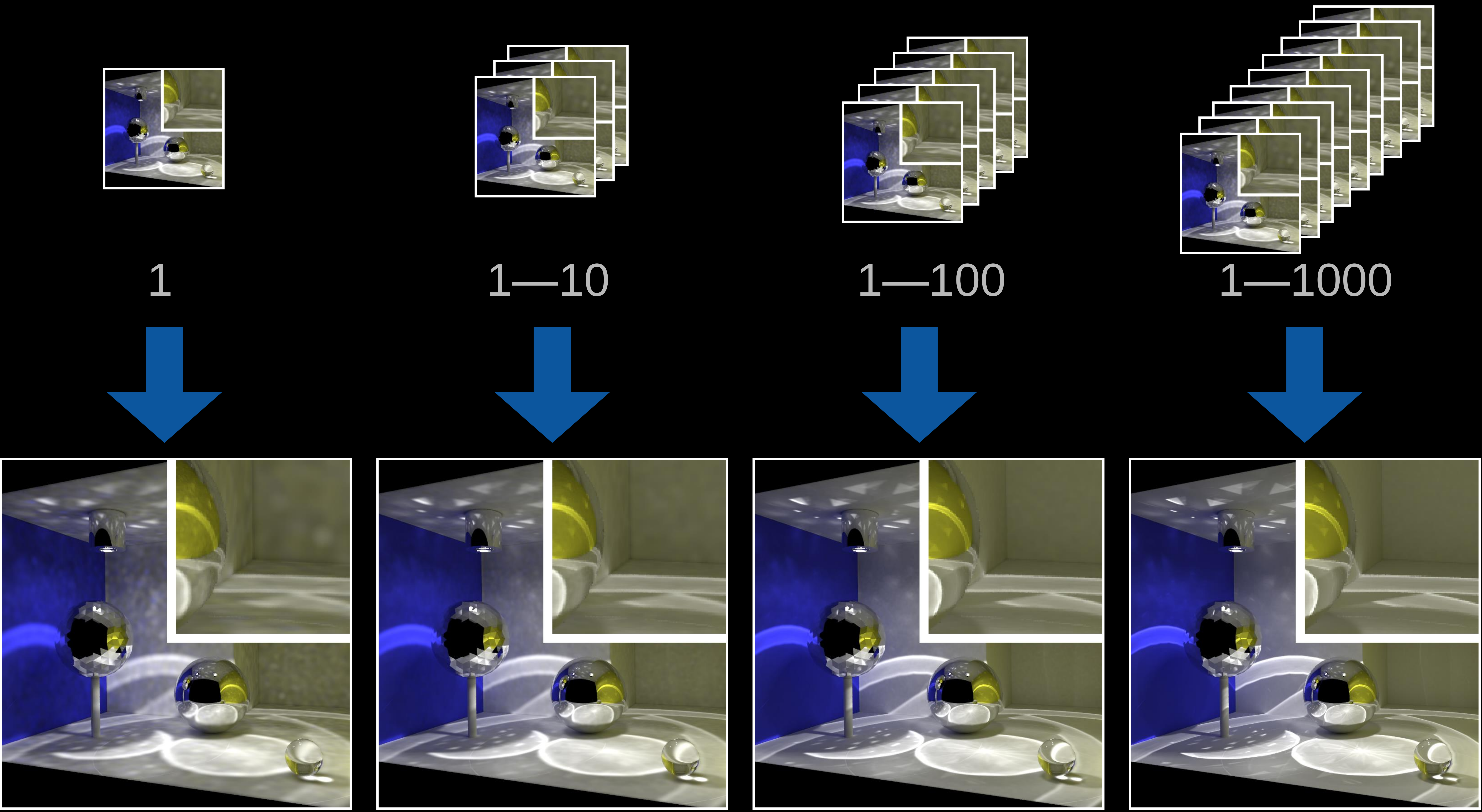
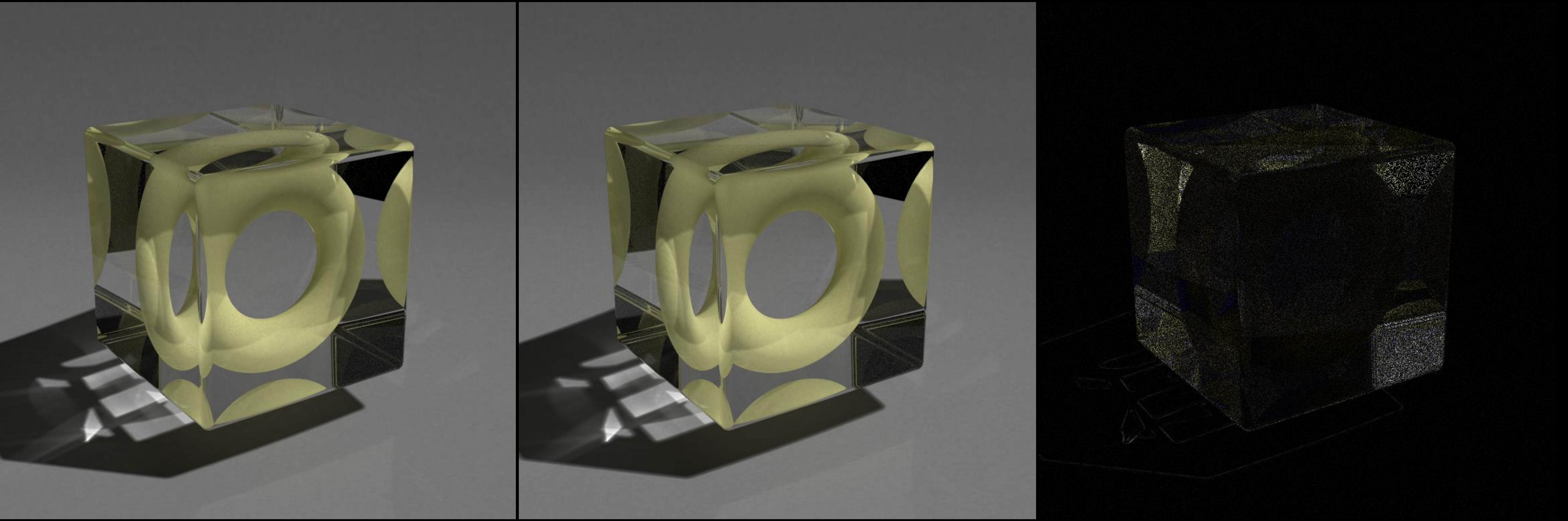


Image 1 Image 10 Image 100 Image 1000

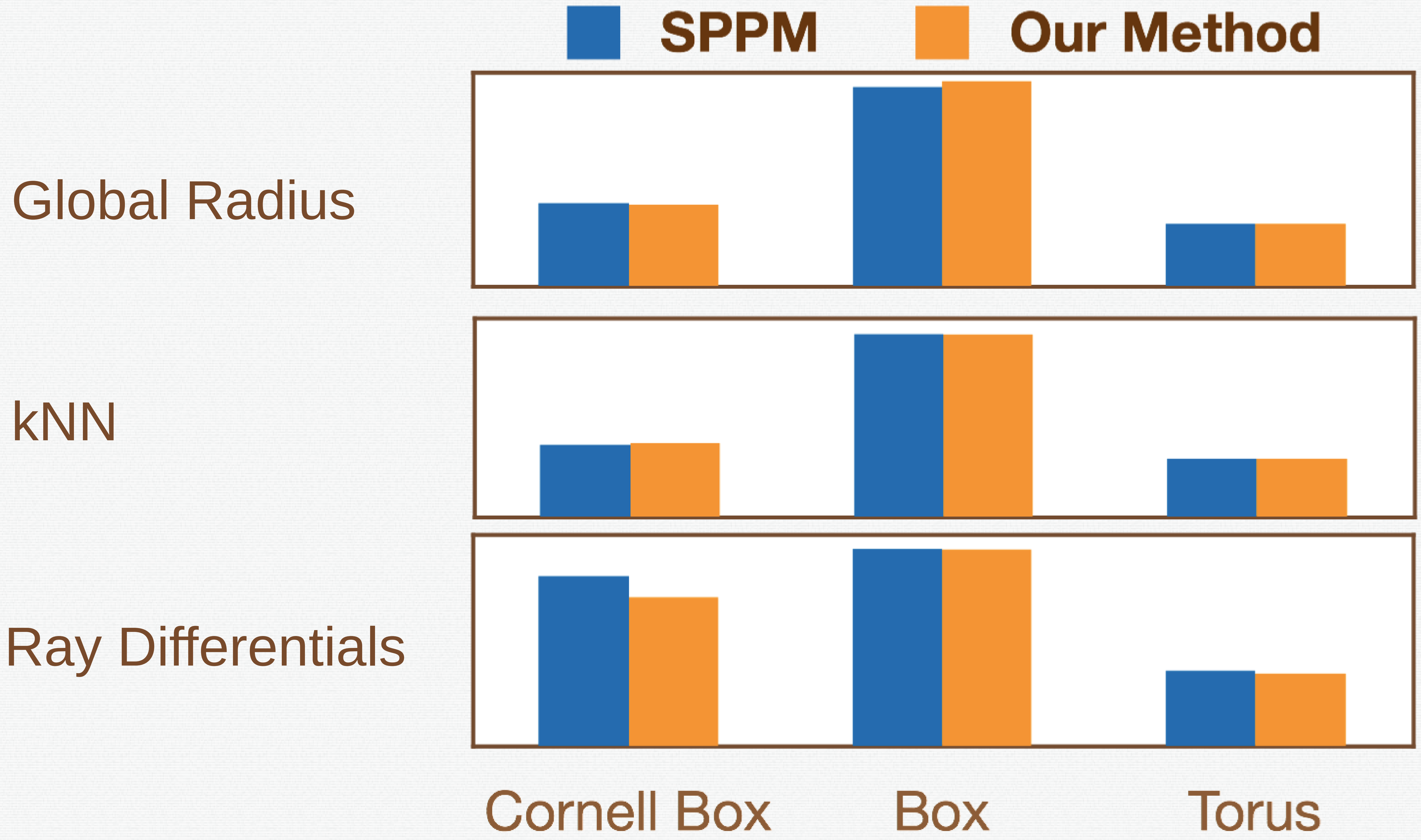

1
1—10
1—100
1—1000

Stochastic PPM | Our method | 20x Difference

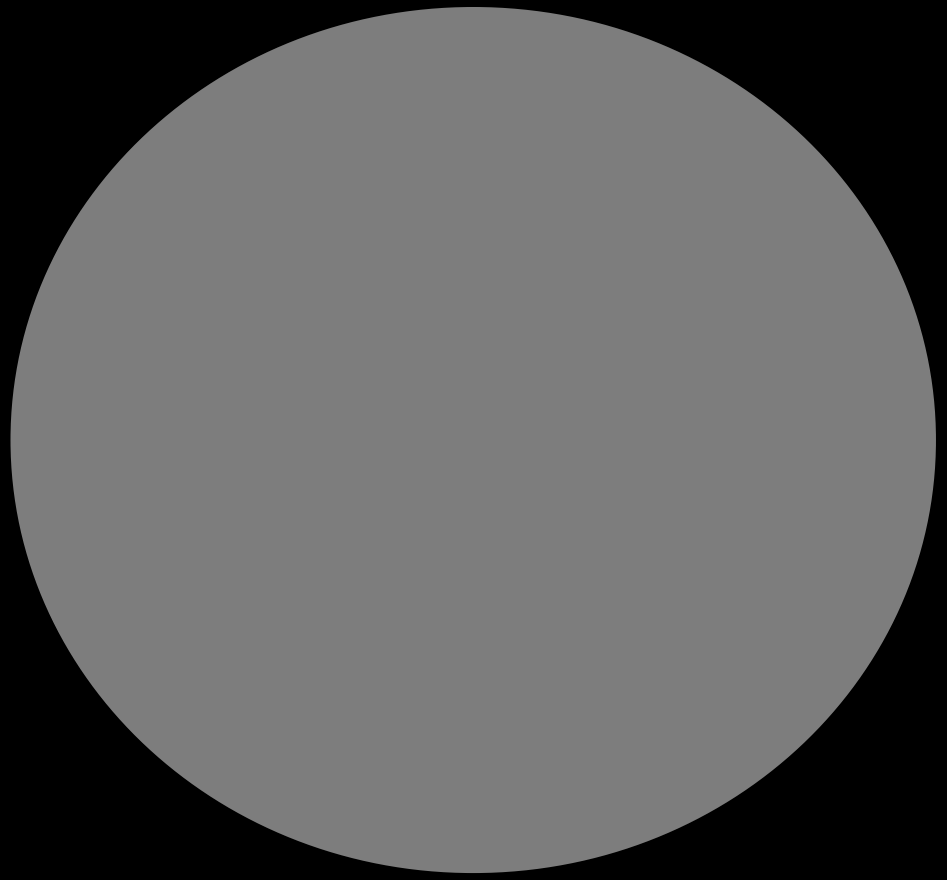


Scene courtesy of Toshiya Hachisuka

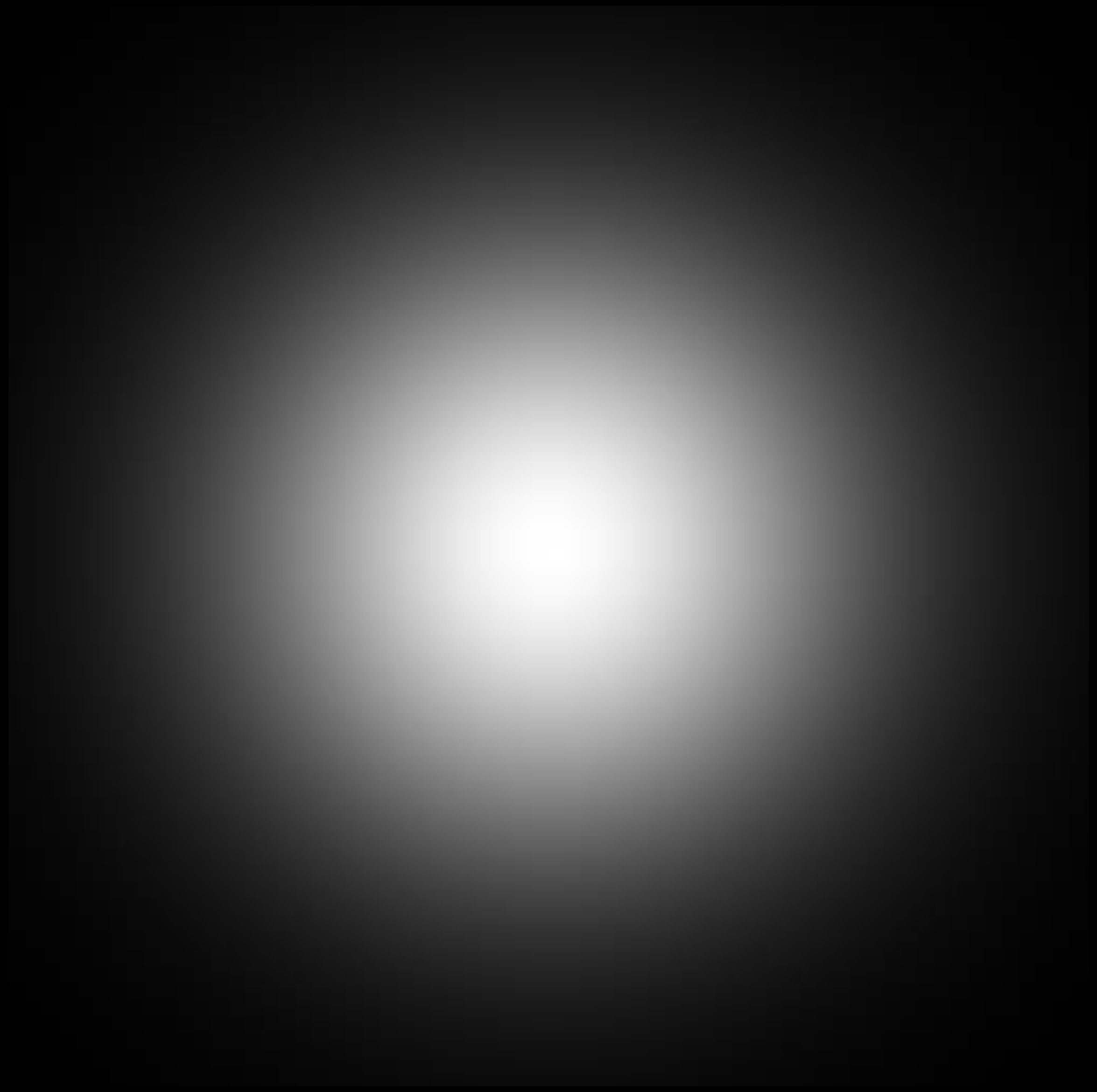
Rendering Time



Arbitrary Kernels

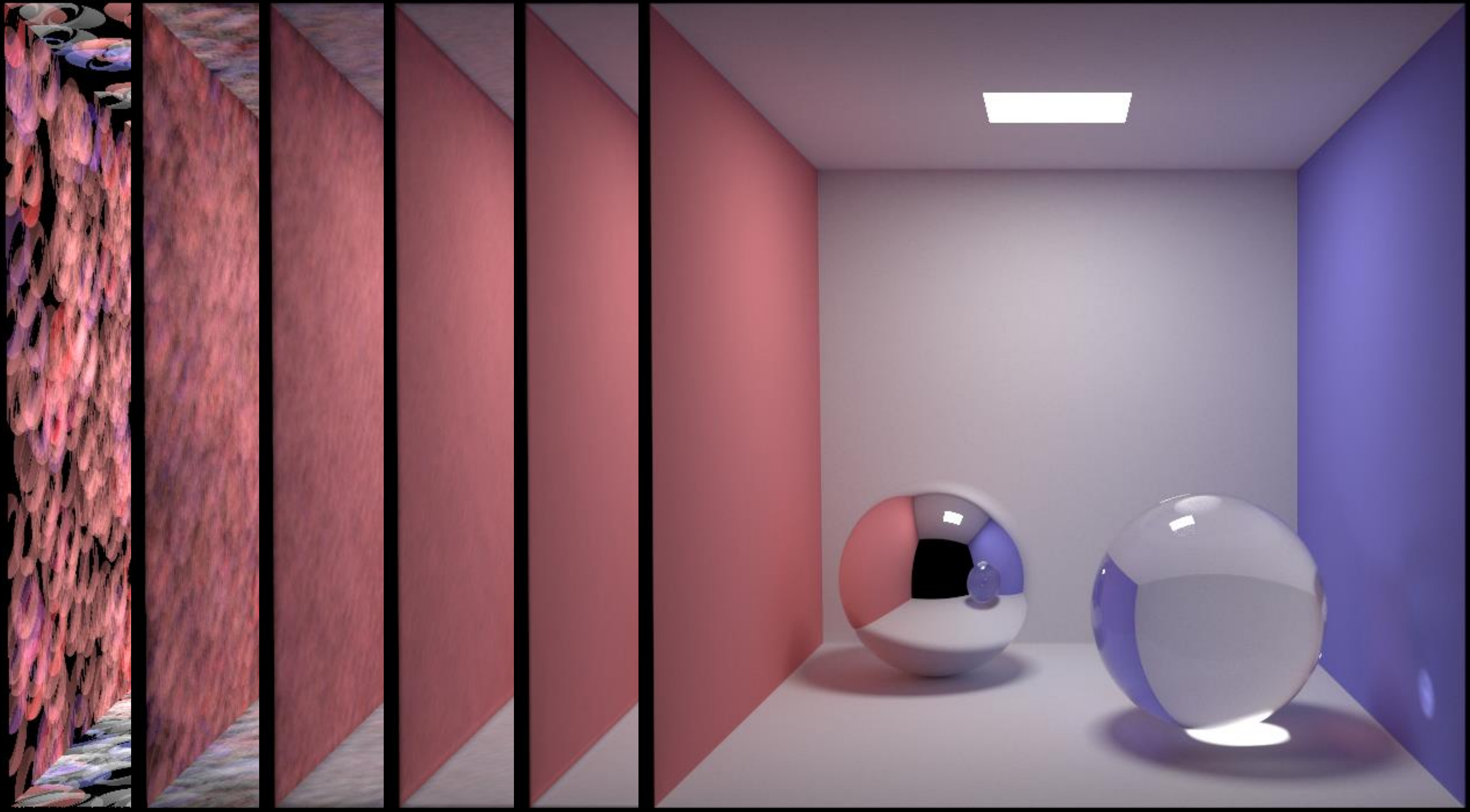


Box

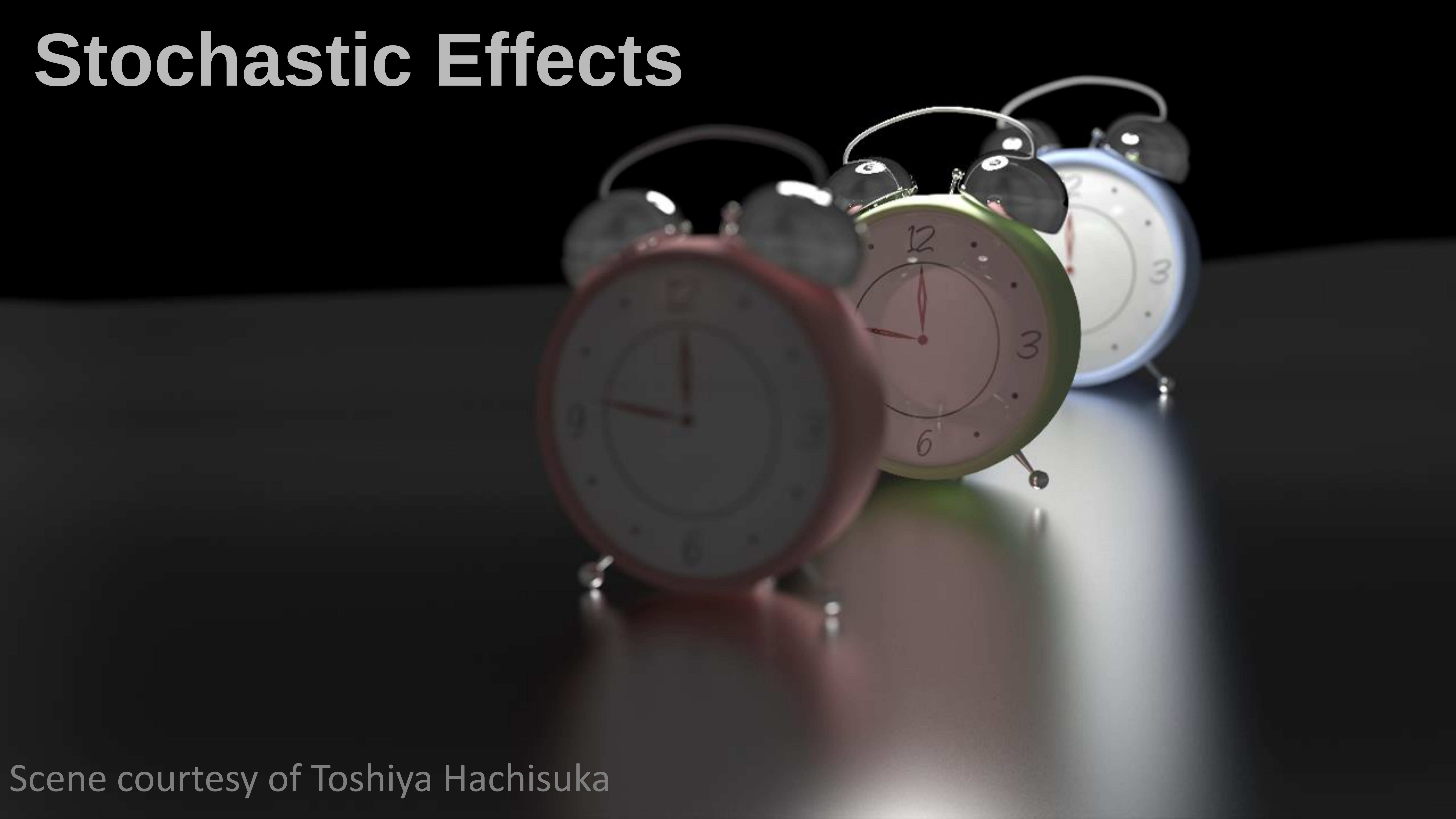


Gaussian

SIGGRAPH



Stochastic Effects



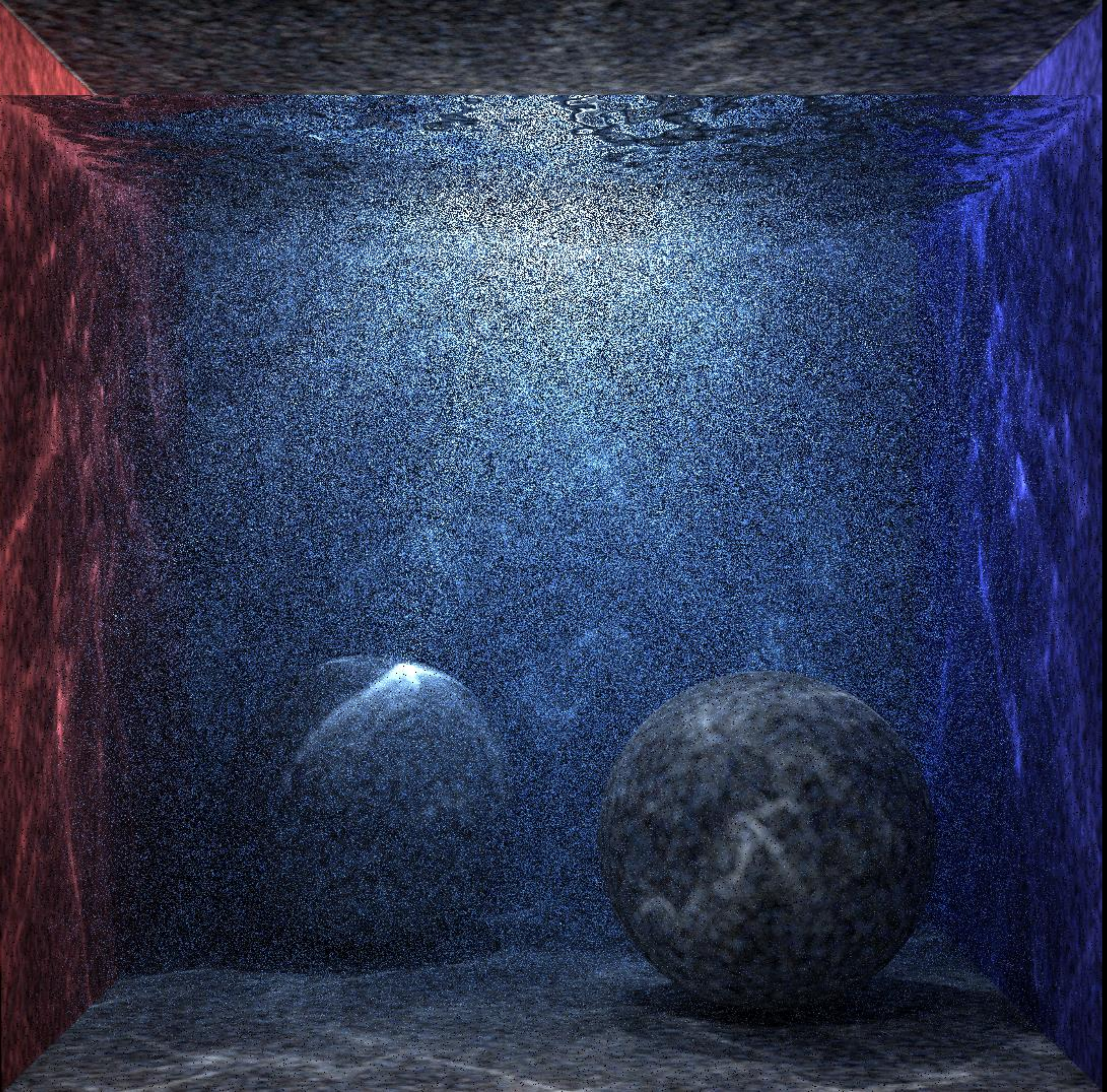
Scene courtesy of Toshiya Hachisuka

Participating Media

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

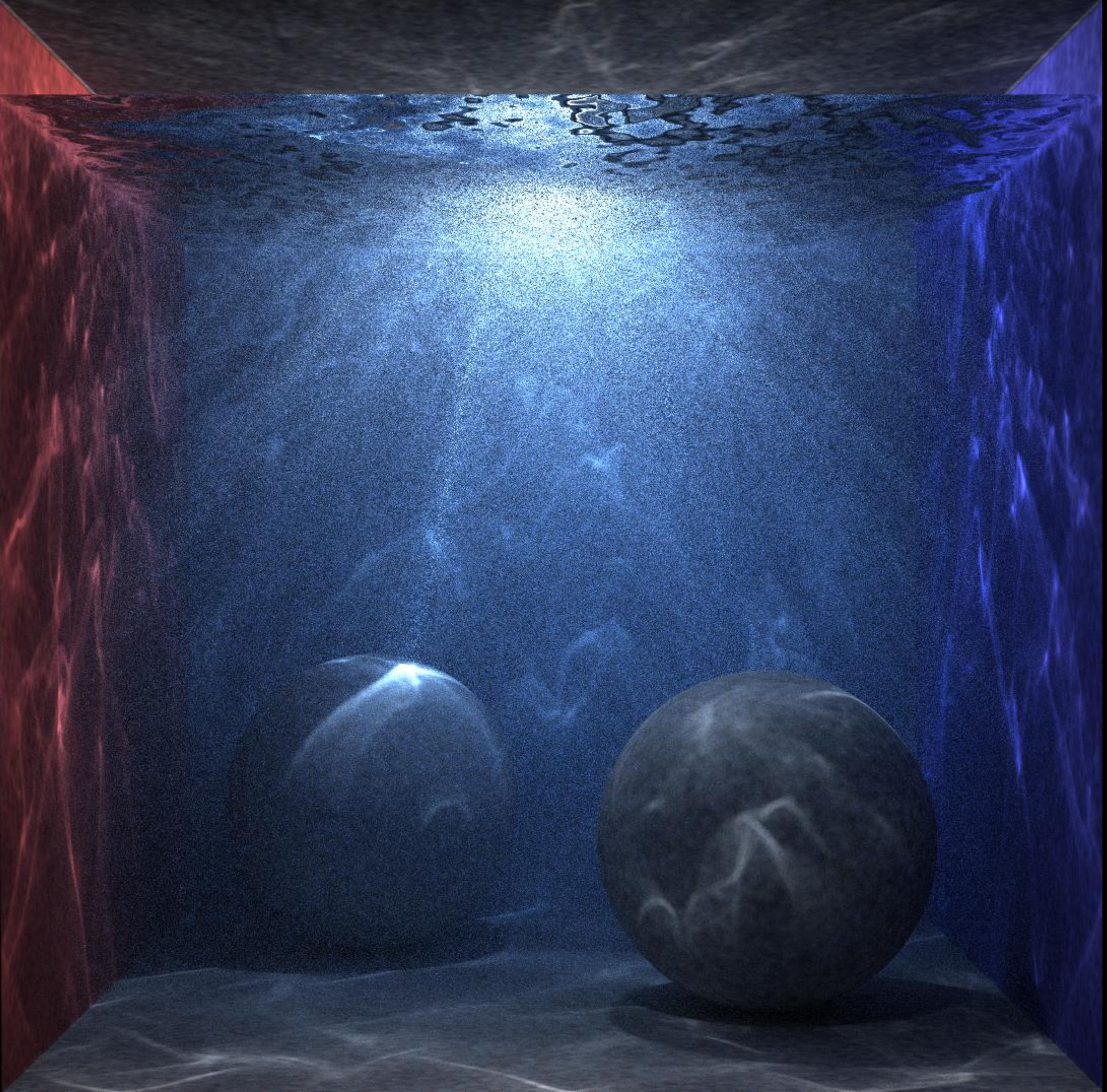
Participating Media

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



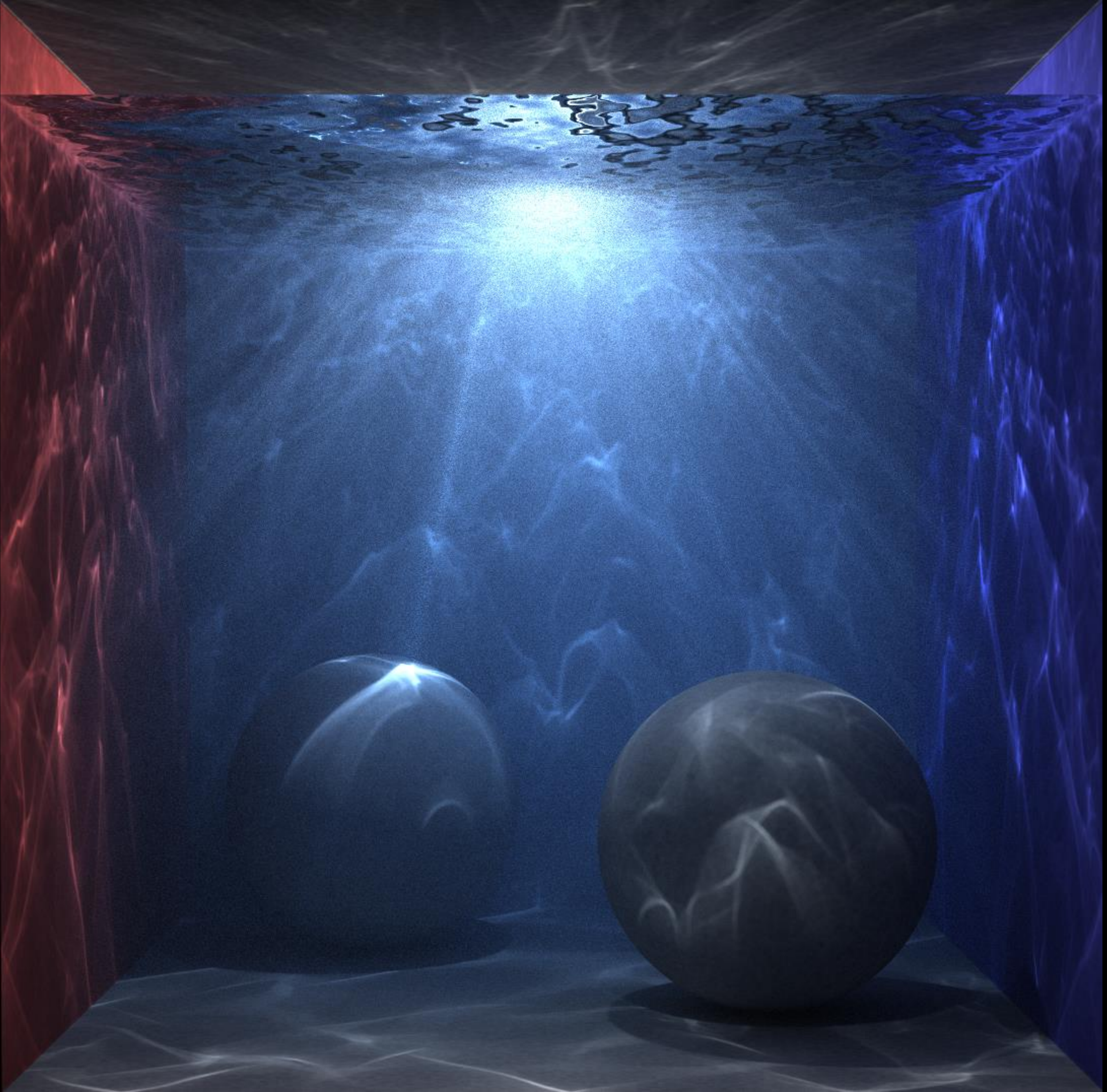
1 iteration

2 million photons



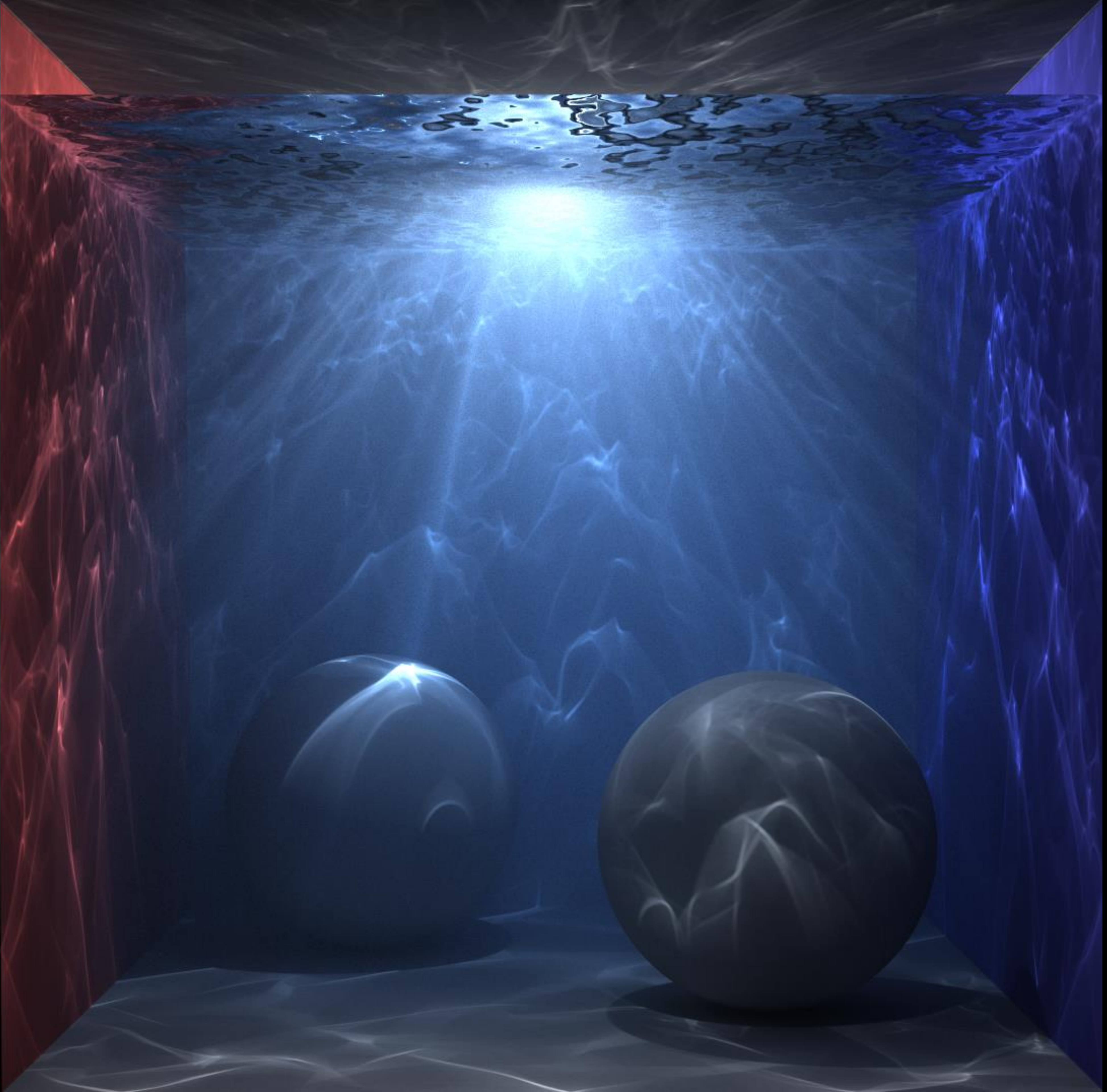
10 iterations

20 million photons



100 iterations

200 million photons



1000 iterations

2 billion photons

Conclusions

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

Acknowledgments

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