Progressive Photon Mapping Extensions

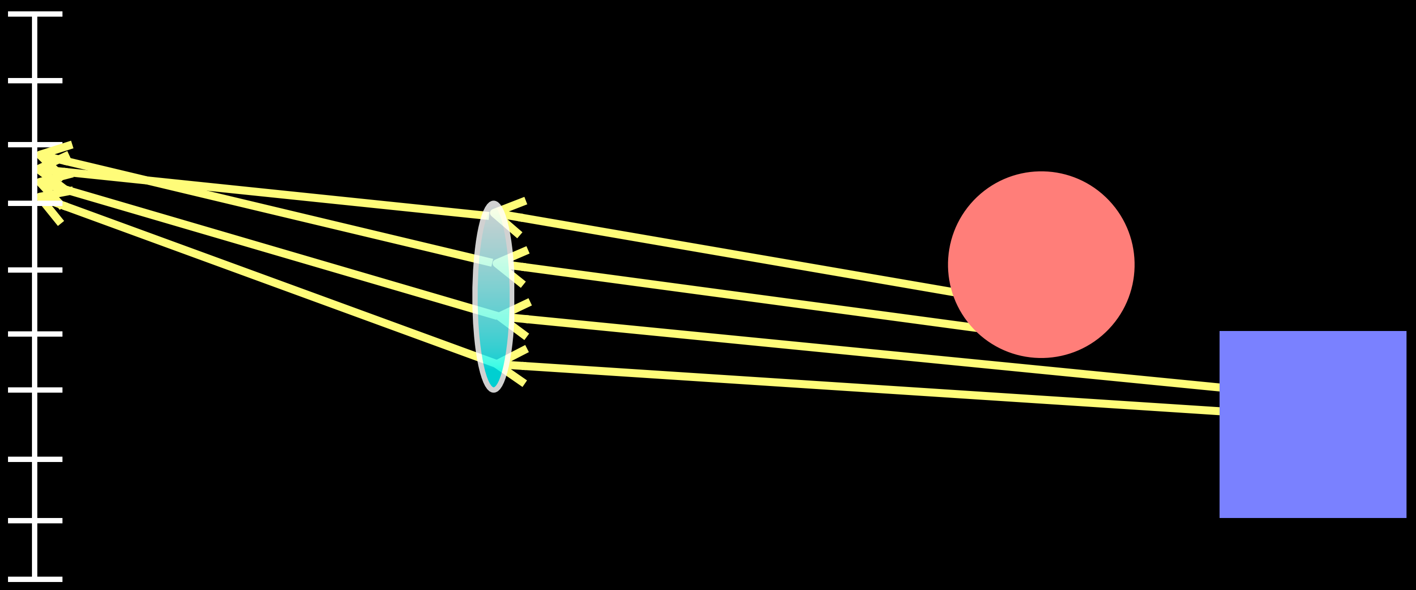
Toshiya Hachisuka
Aarhus University

State of the Art in Photon Density Estimation

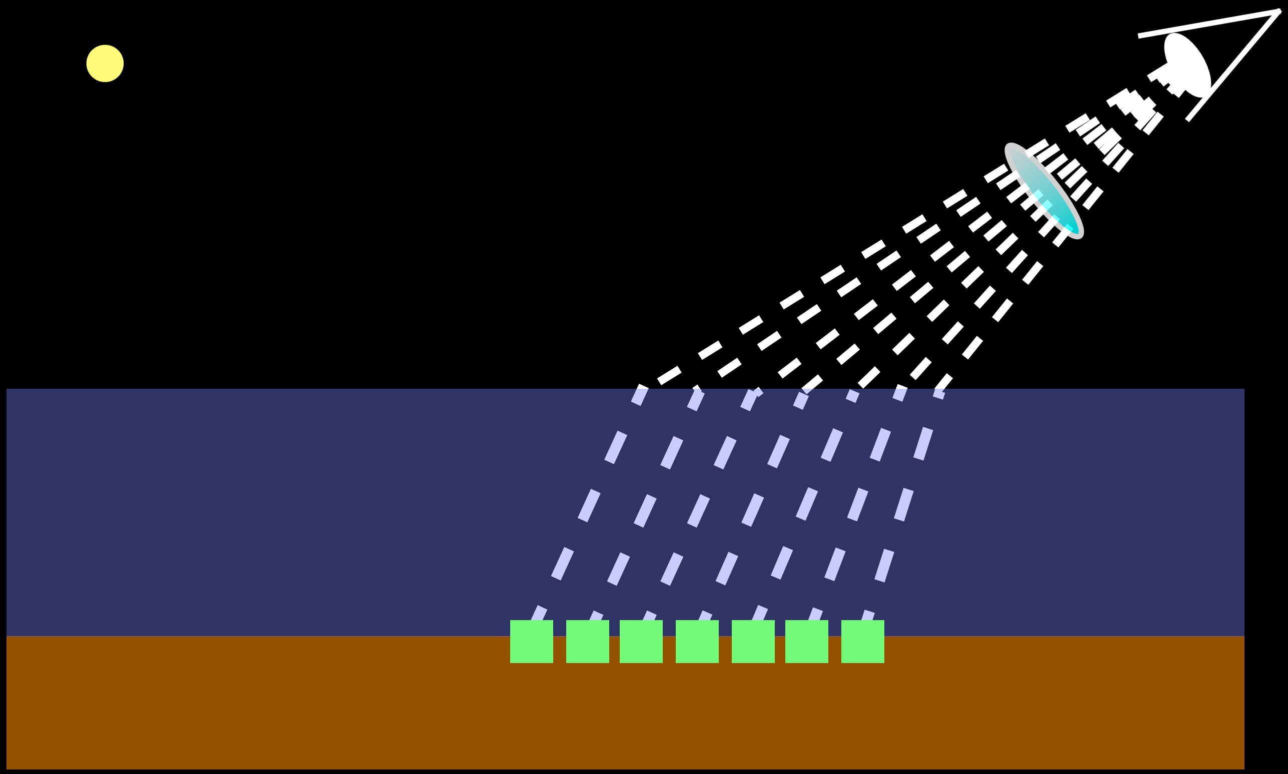

SANDY
contractors
594461

Distribution Ray Tracing

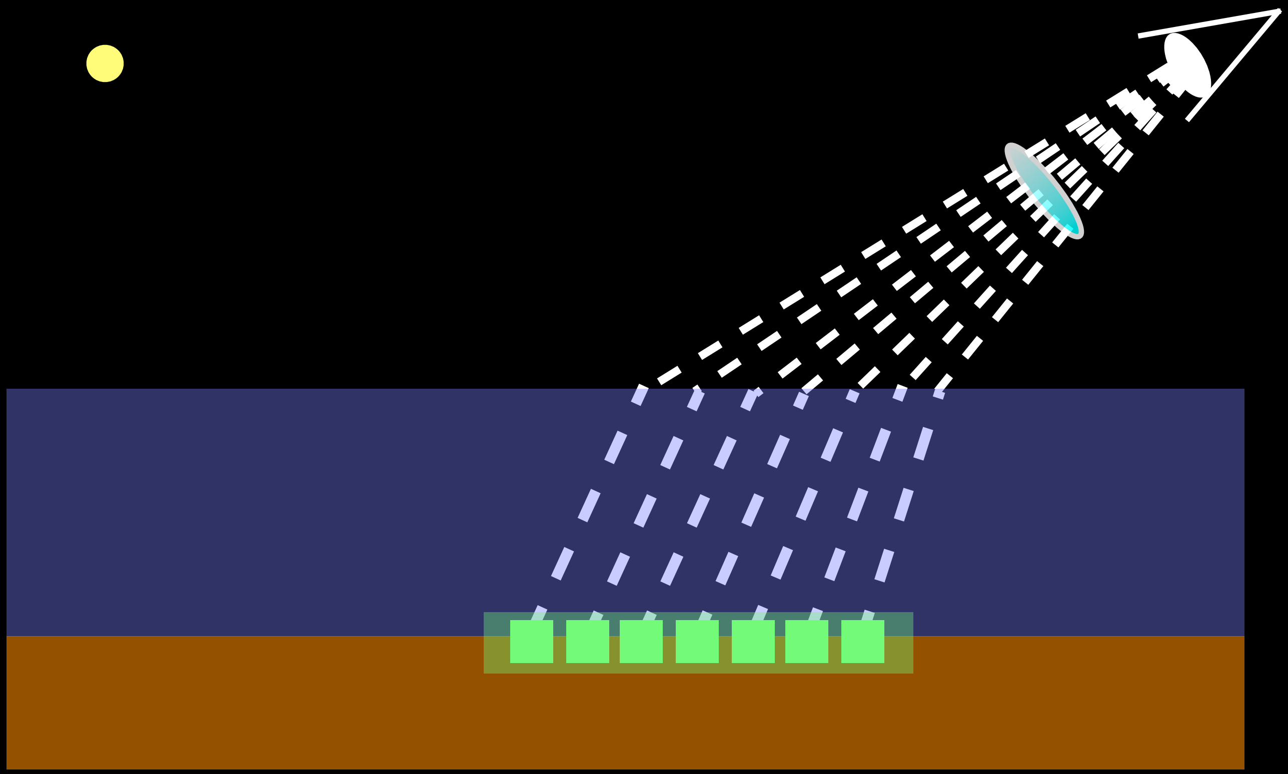
- Computes average illumination [Cook et al. 84]



Lens Simulation with PPM

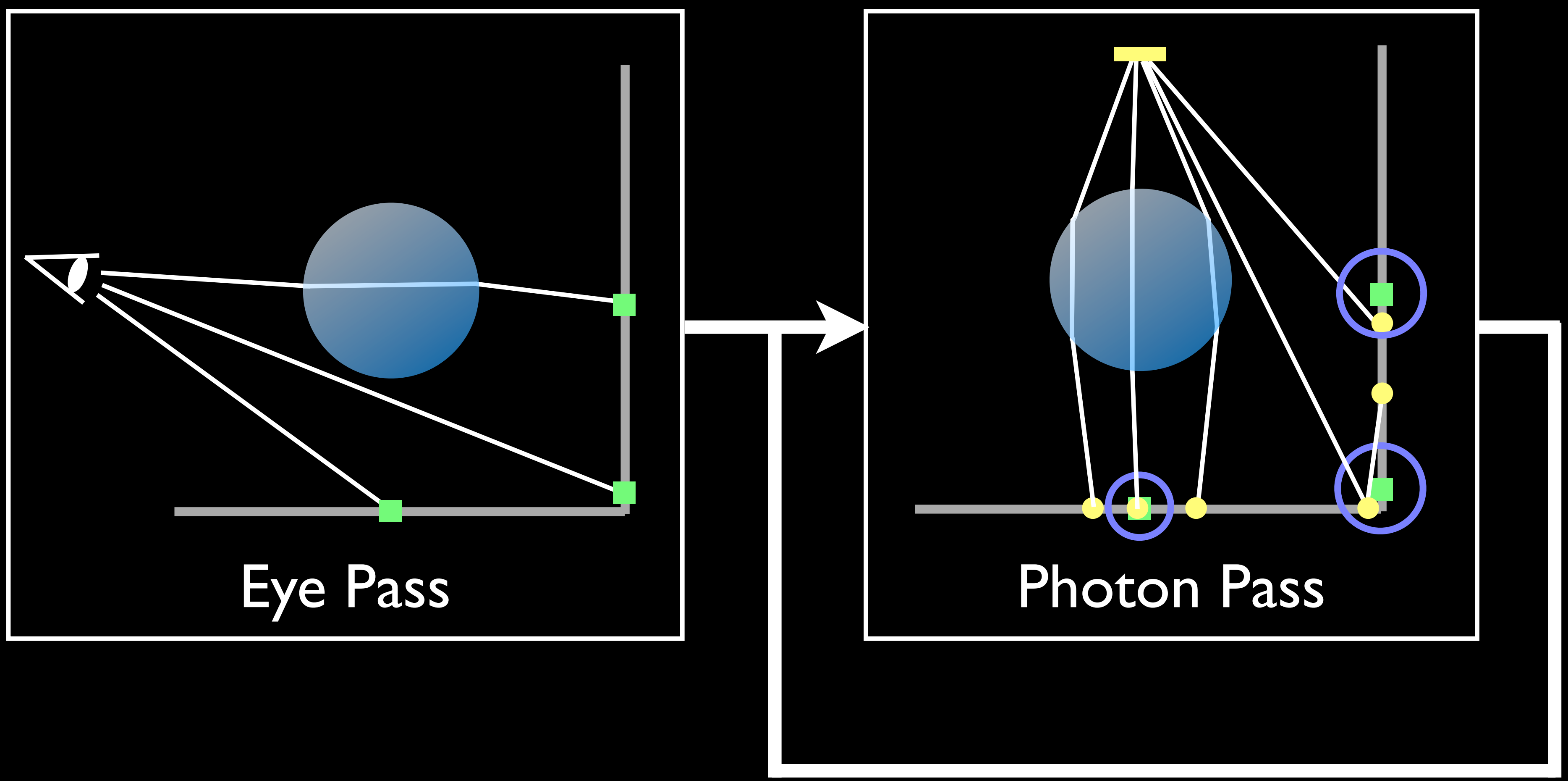


Lens Simulation with PPM

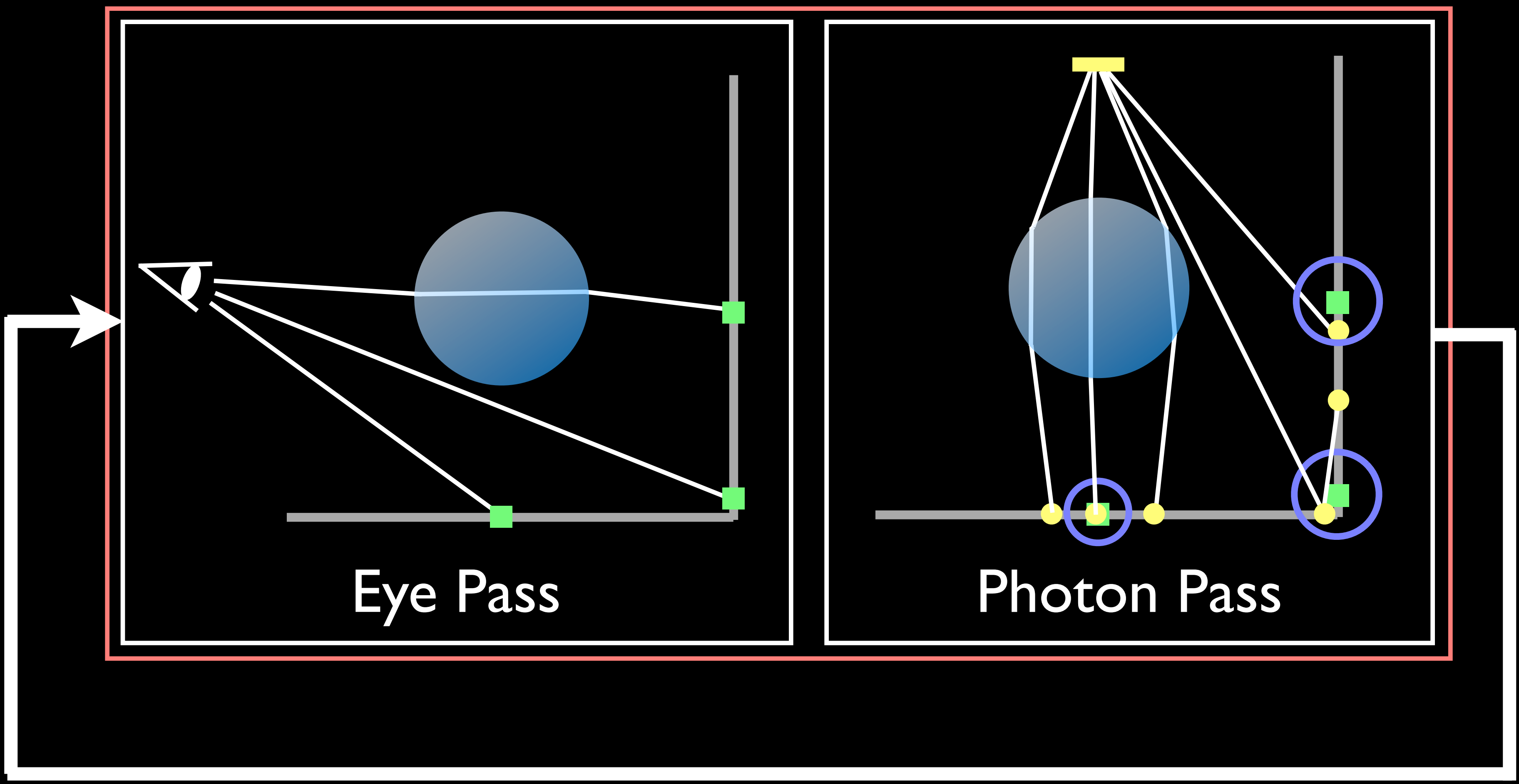


Infinite number of measurement points

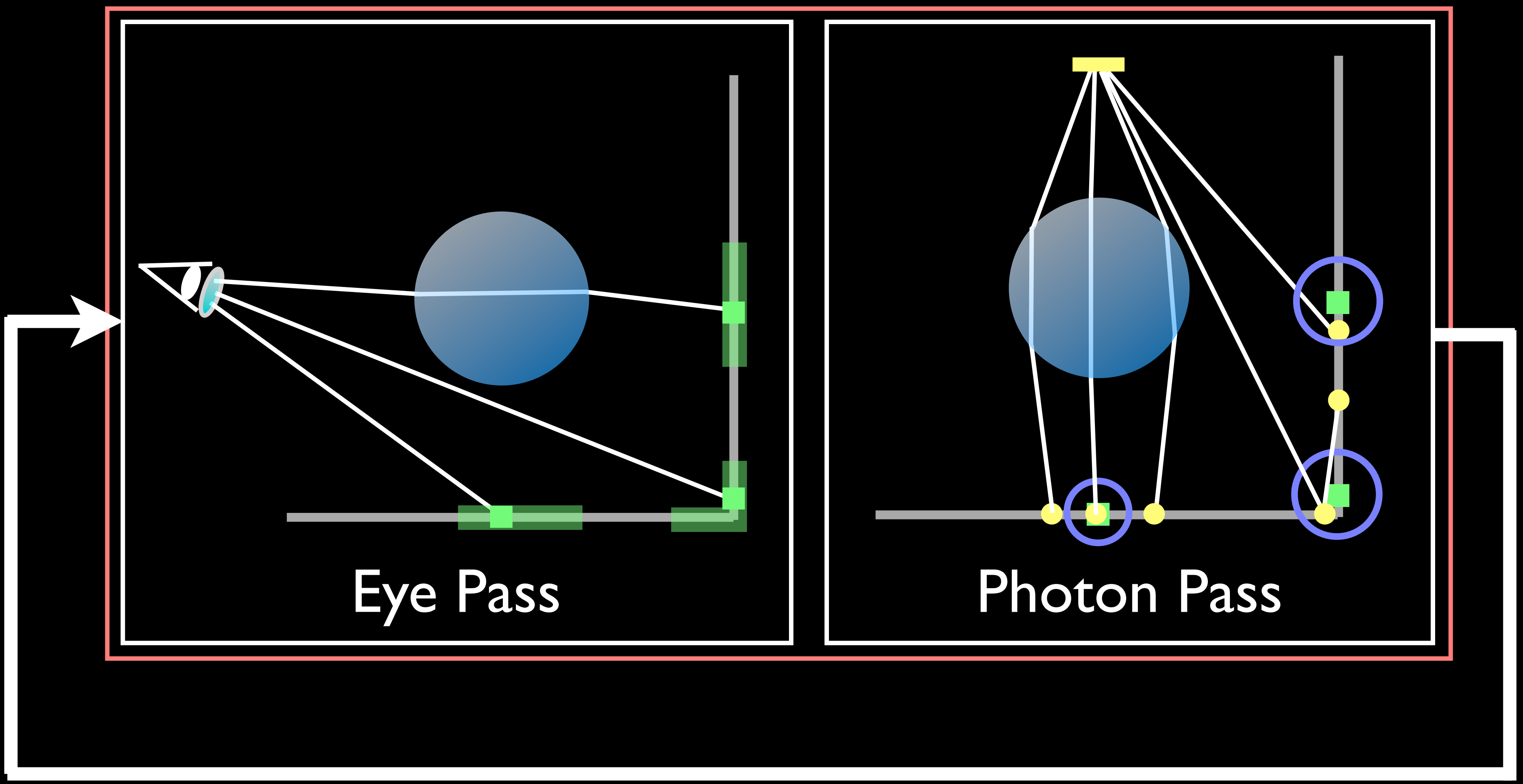
PPM



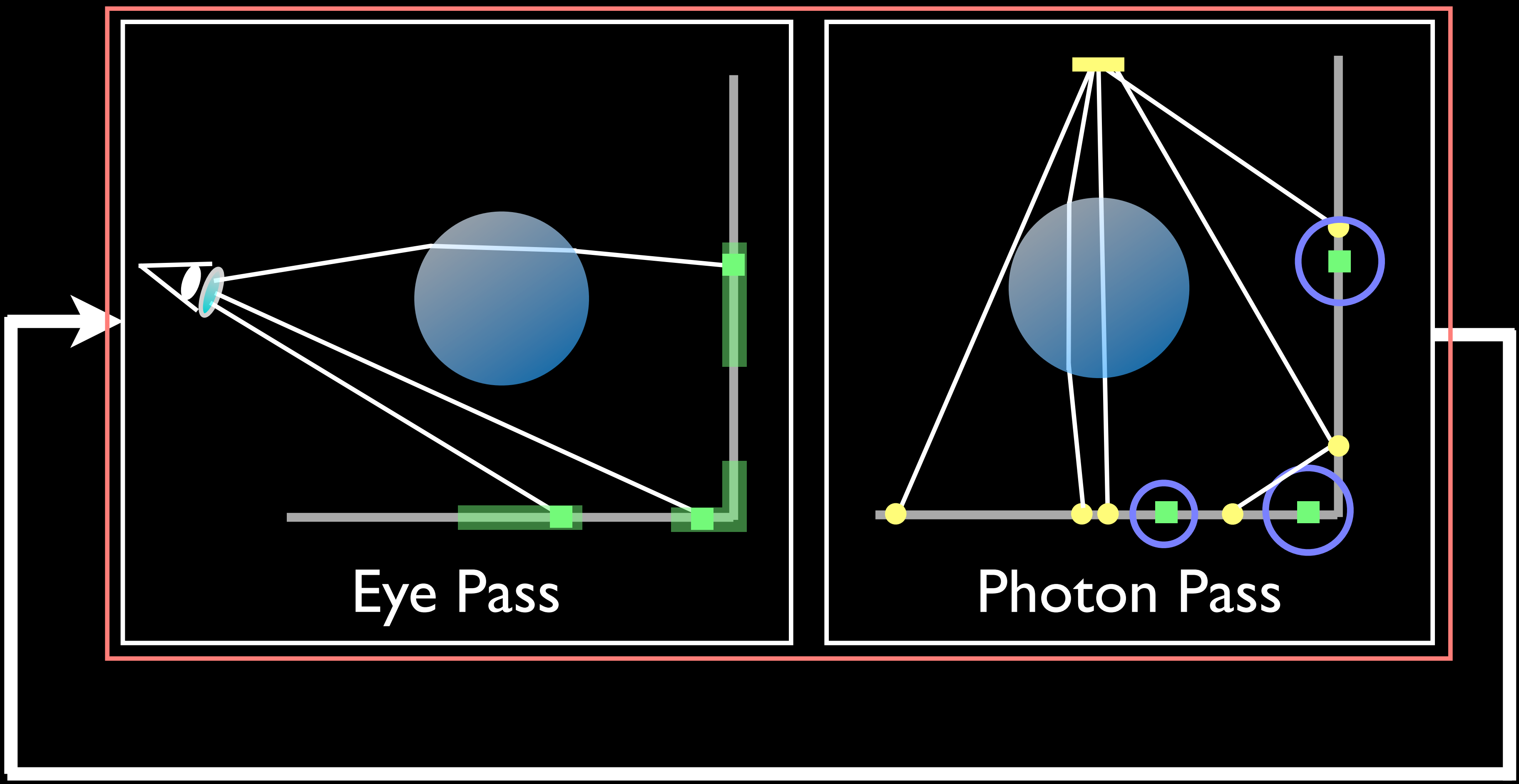
Stochastic PPM



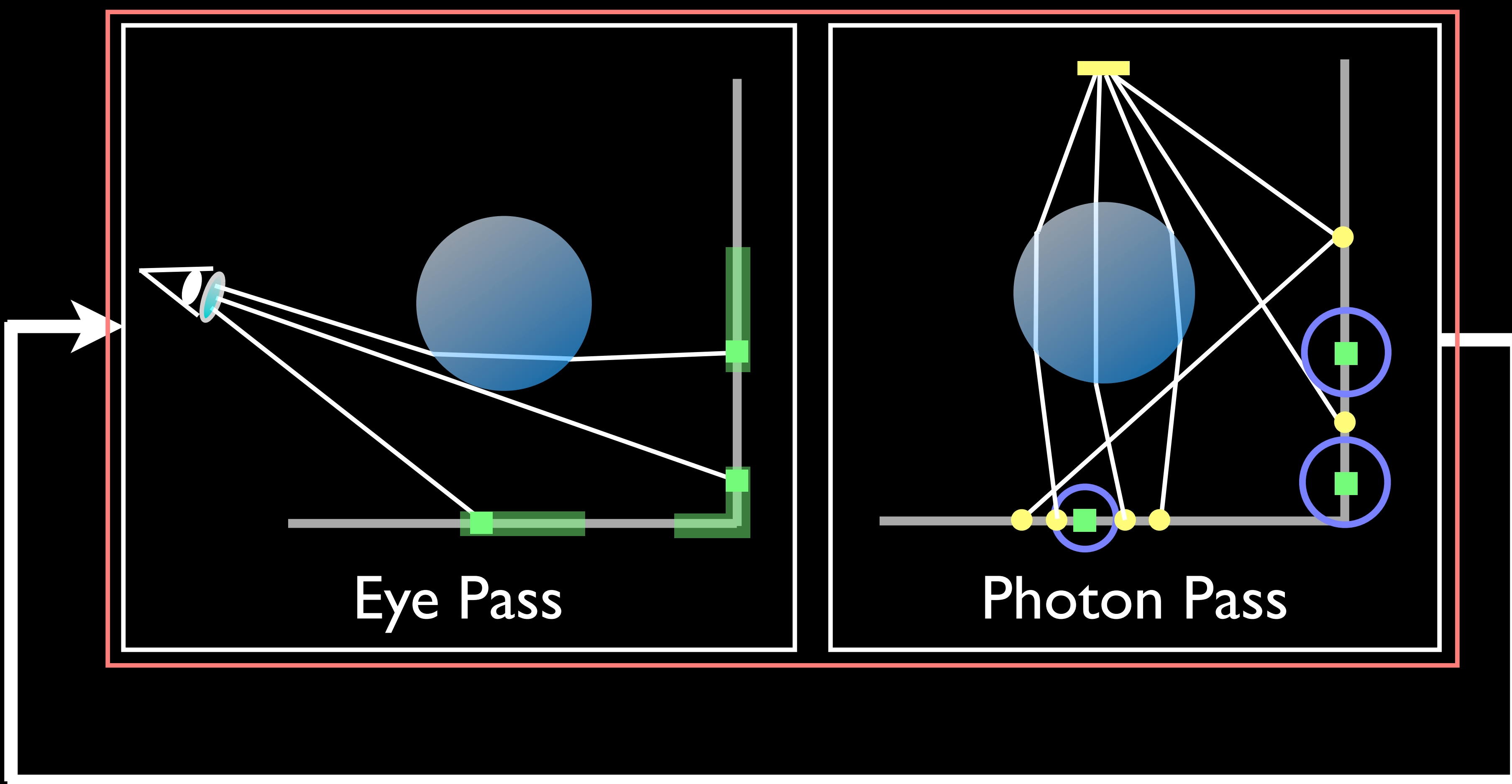
Stochastic PPM



Stochastic PPM



Stochastic PPM



Stochastic Progressive Density Estimation

$$L_i\left(S,\vec{\omega}\right) = \frac{\tau_i\left(S,\vec{\omega}\right)}{\pi R_i\left(S\right)^2 N_e(i)}$$

$$\boxed{\lim_{i\to\infty} L_i\left(S,\vec{\omega}\right) = L\left(S,\vec{\omega}\right)}$$

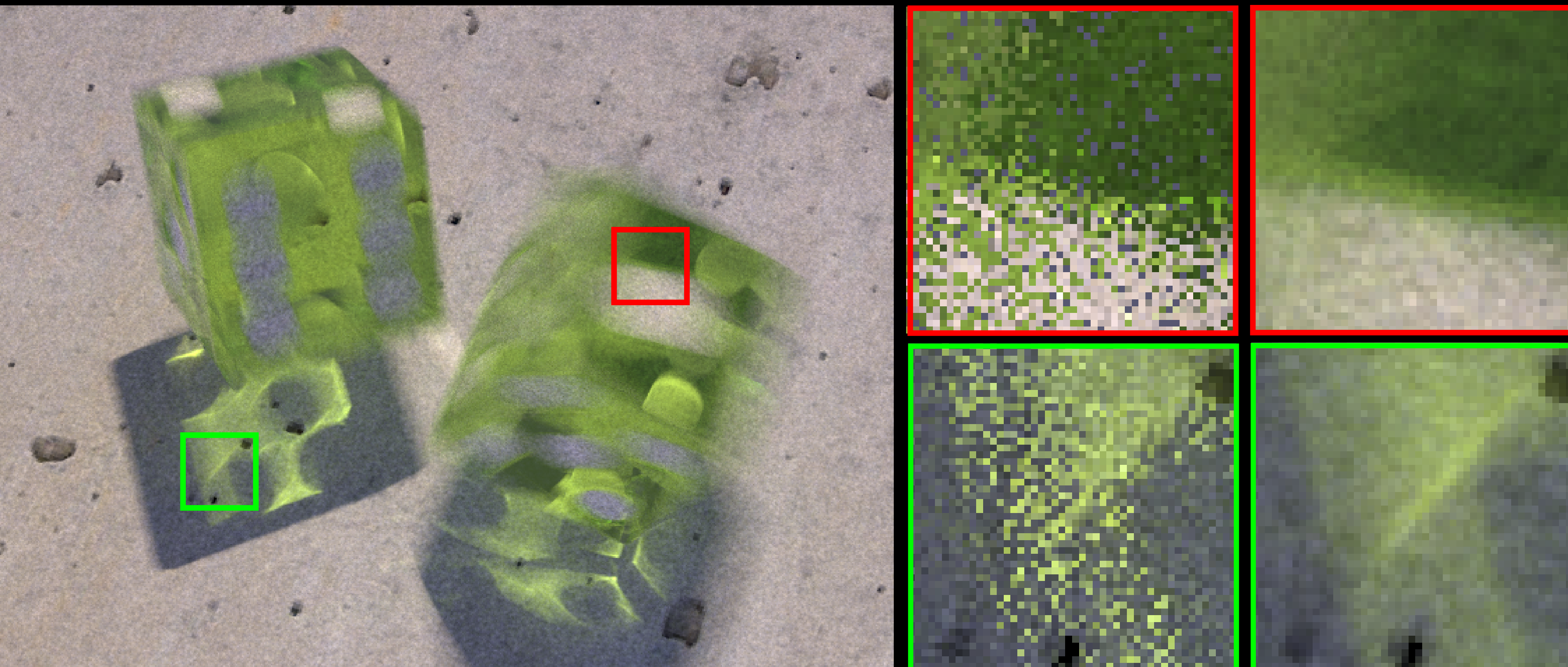
Provable convergence to
average photon density over a region S

Bidirectional Path Tracing

Original PPM

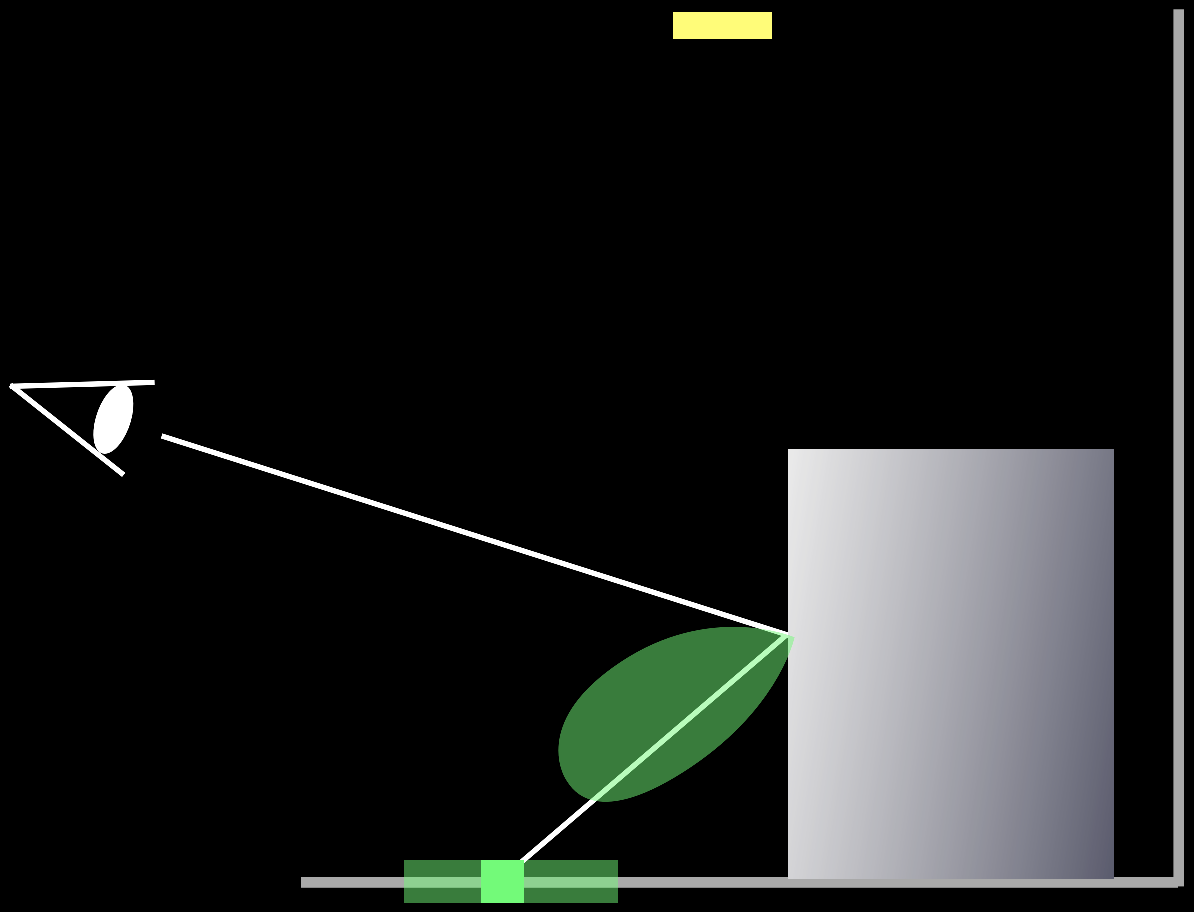
Stochastic PPM

Motion Blur



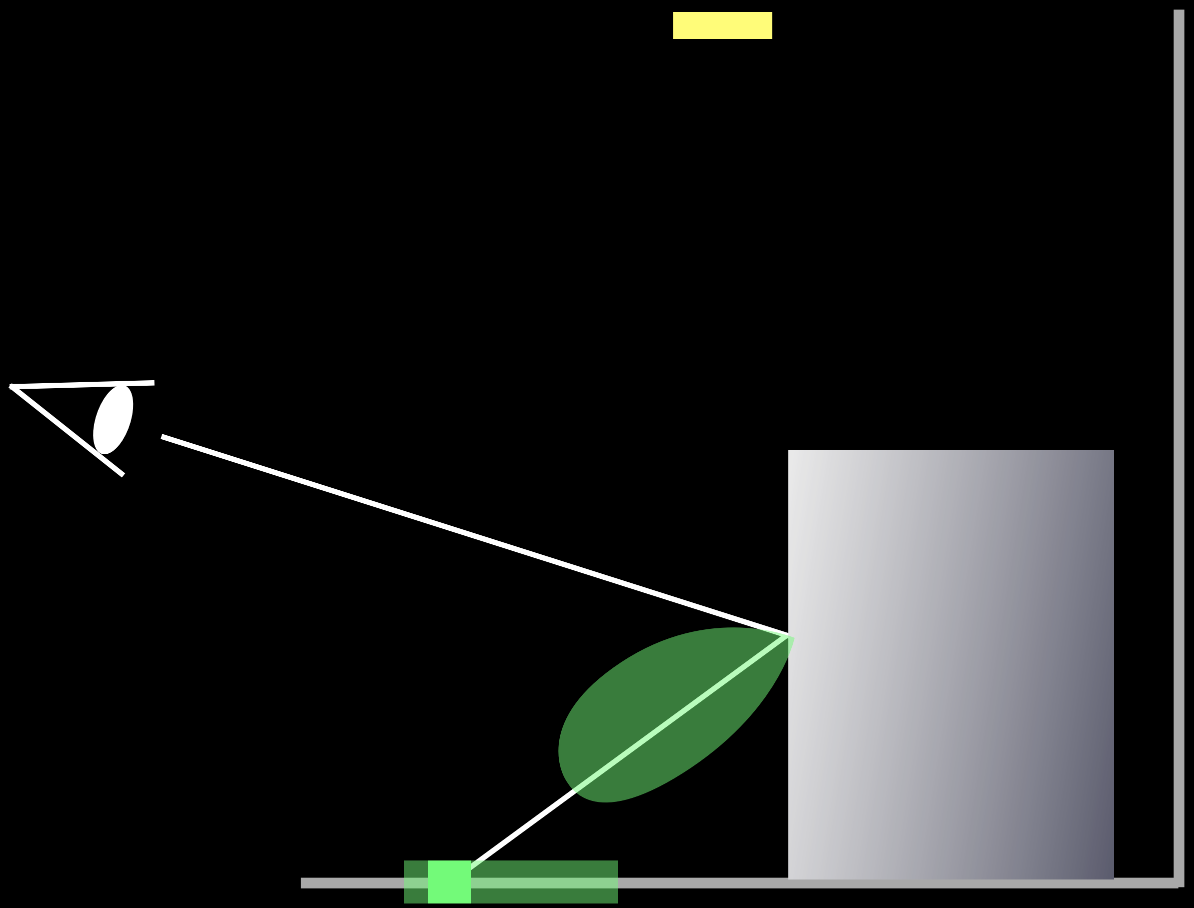
Equal time, Equal memory PPM SPPM

Glossy Materials with SPPM



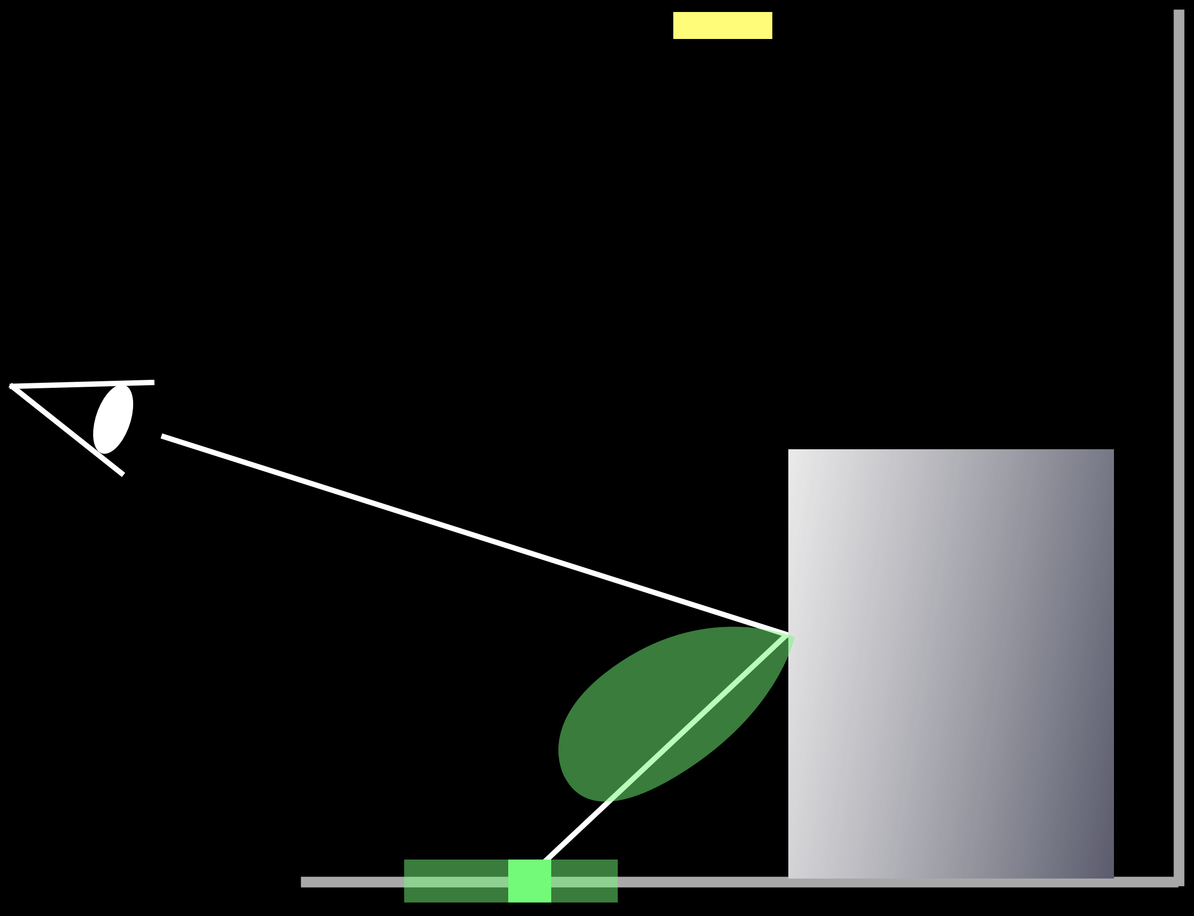
Trace one bounce rays

Glossy Materials with SPPM



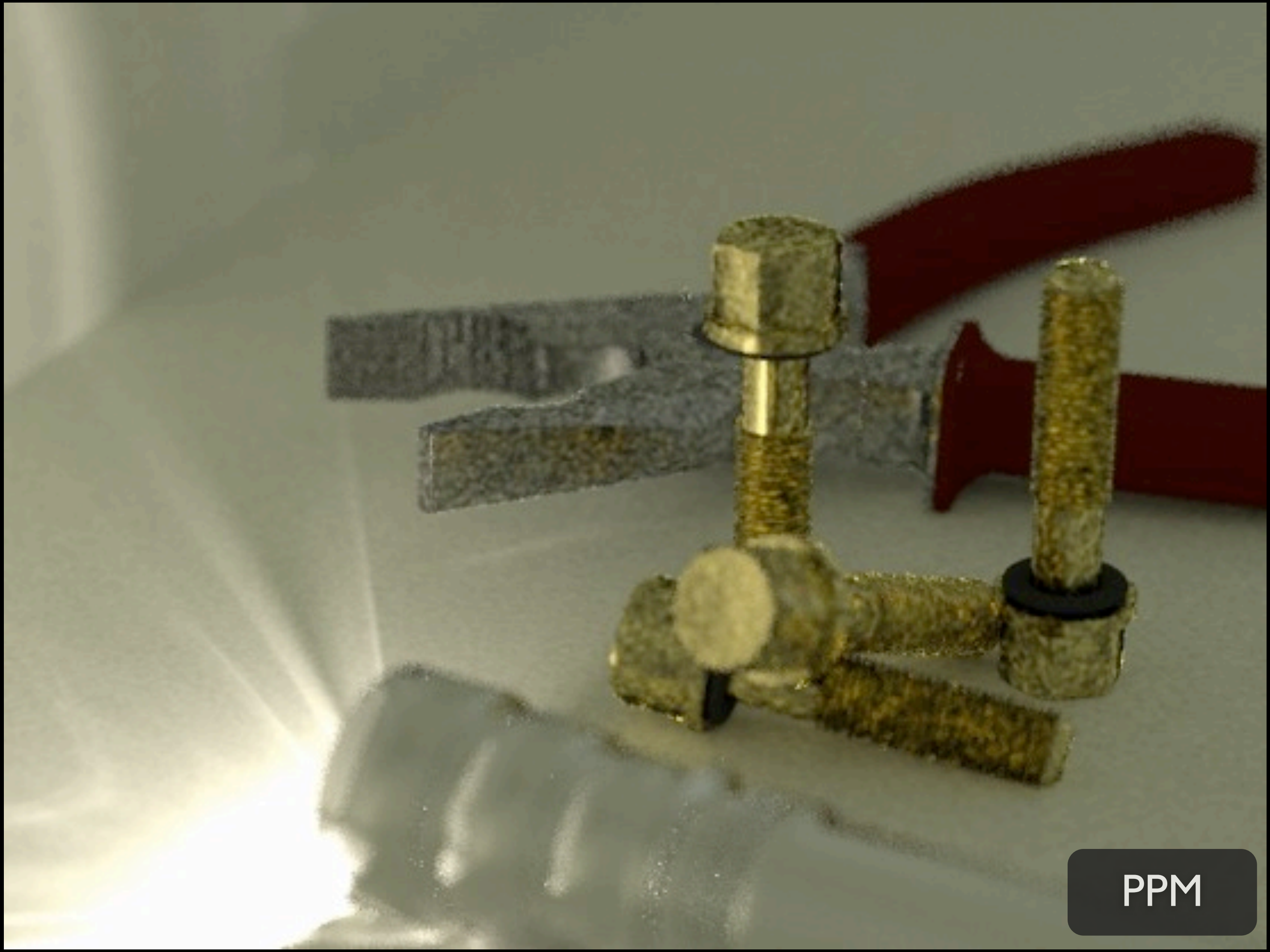
Trace one bounce rays

Glossy Materials with SPPM

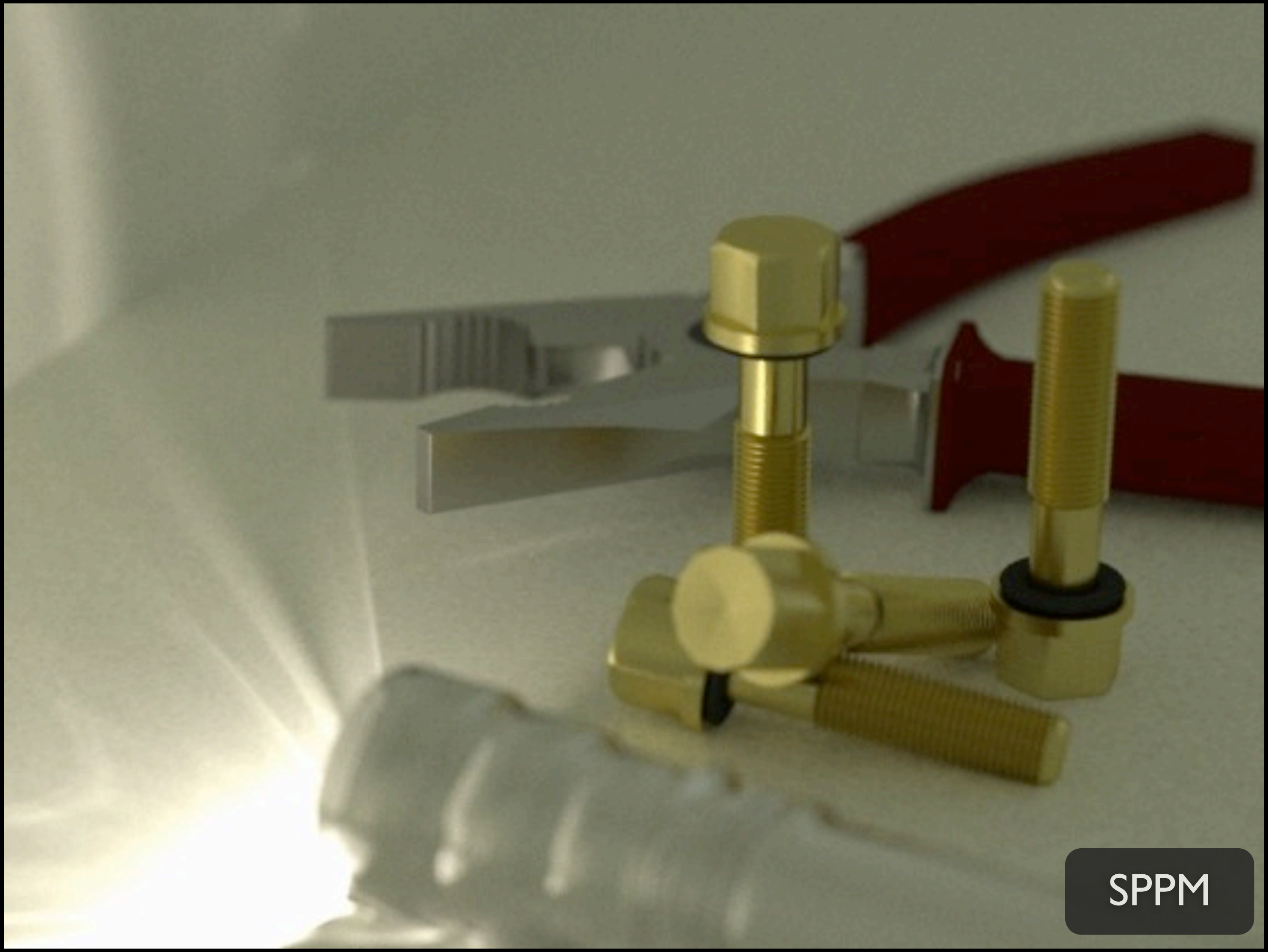


Trace one bounce rays

DOF + Glossy Reflection + Caustics

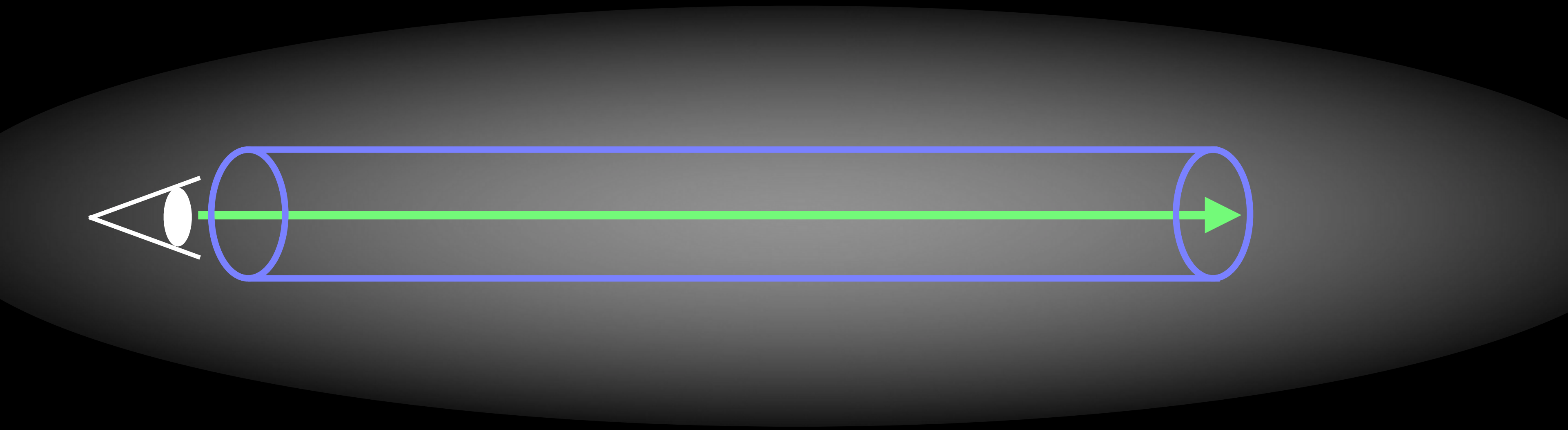


DOF + Glossy Reflection + Caustics



Participating Media

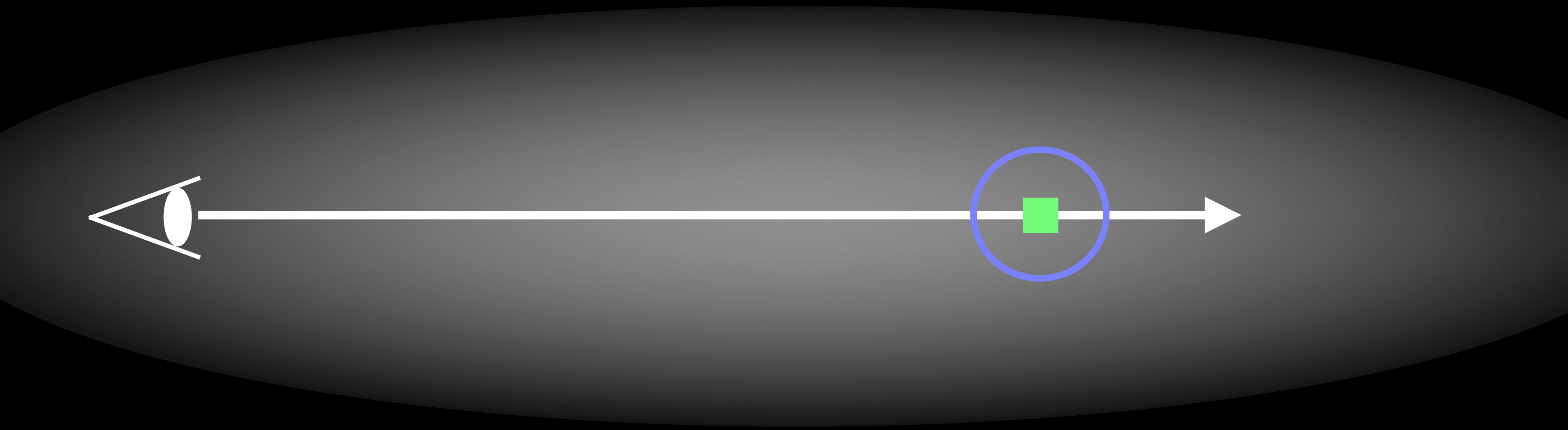
- Two basic approaches
  - Progressive beam radiance estimate (PPM)
  - Stochastically sample a point along eye ray (SPPM)



PPM style: cylinder progressive estimate

Participating Media

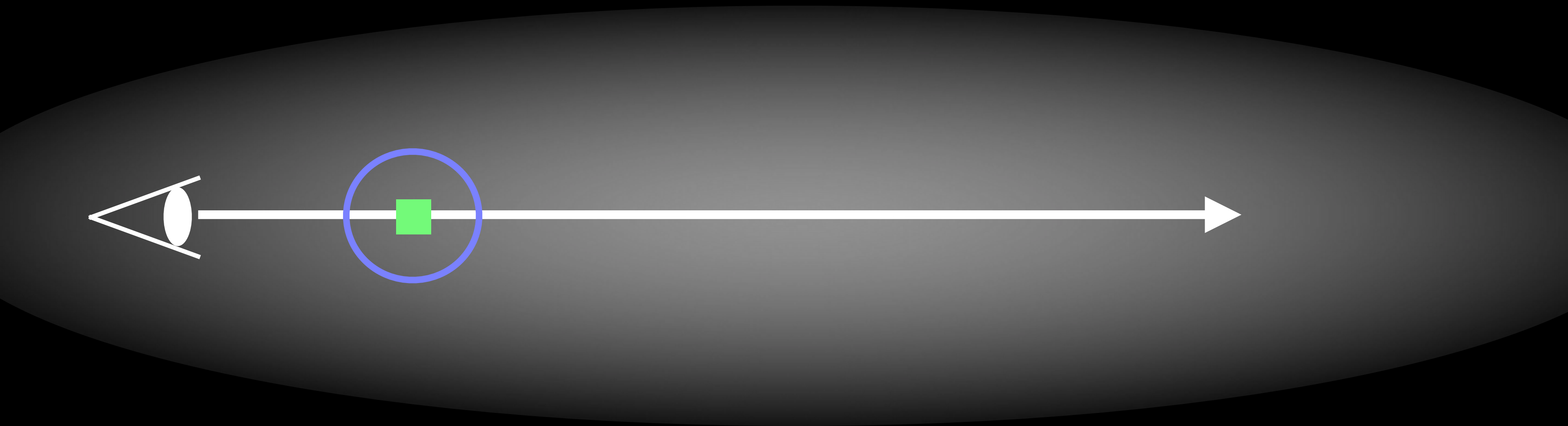
- Two basic approaches
  - Progressive beam radiance estimate (PPM)
  - Stochastically sample a point along eye ray (SPPM)



SPPM style: stochastic sampling along a ray

Participating Media

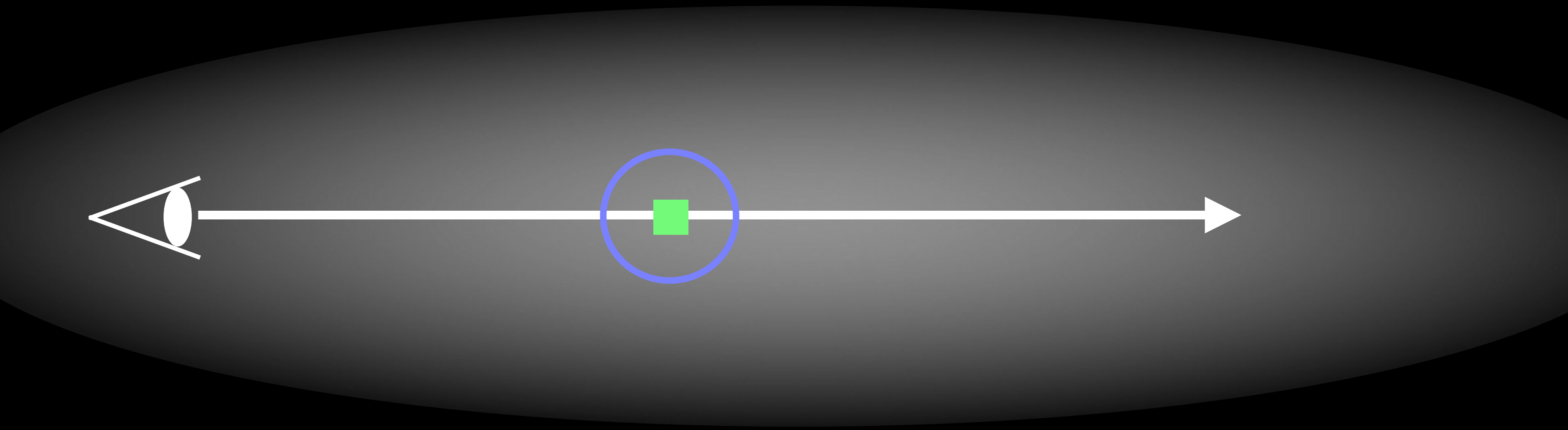
- Two basic approaches
  - Progressive beam radiance estimate (PPM)
  - Stochastically sample a point along eye ray (SPPM)



SPPM style: stochastic sampling along a ray

Participating Media

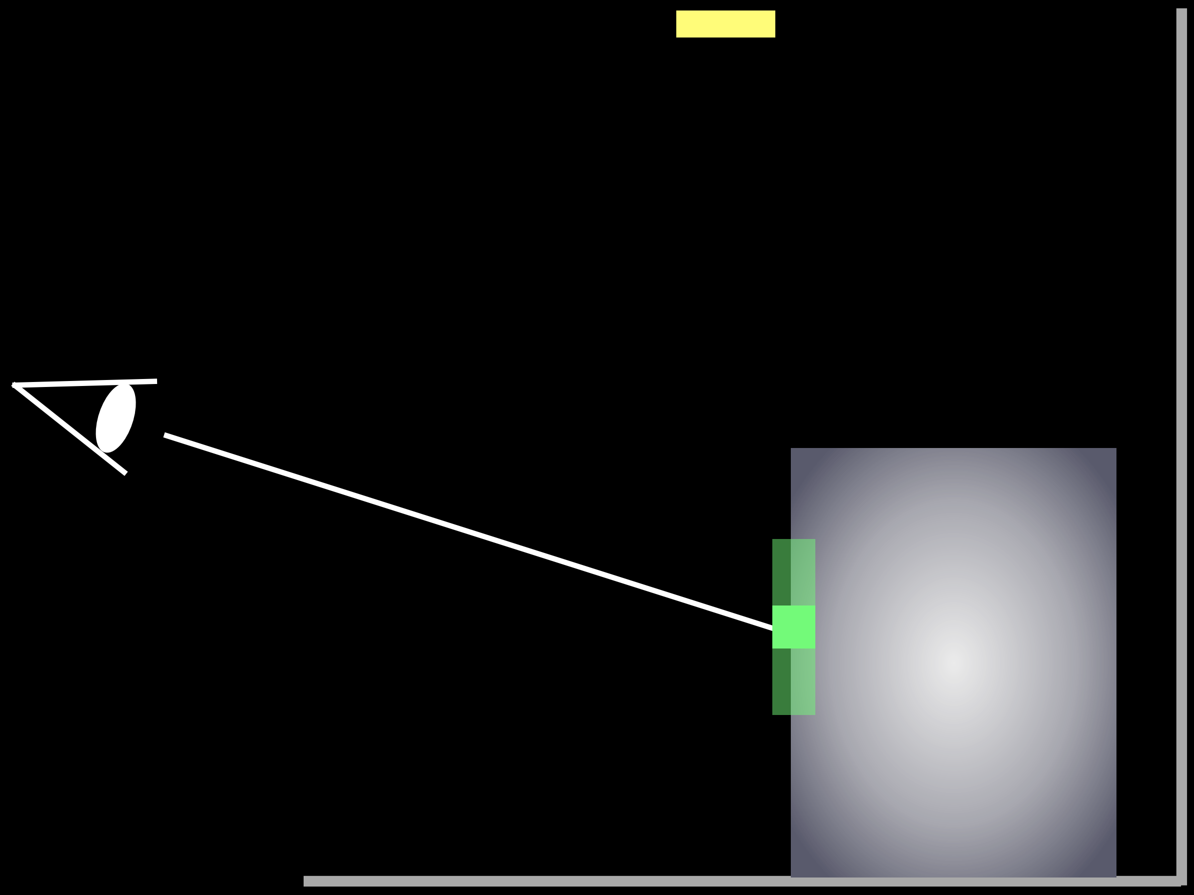
- Two basic approaches
  - Progressive beam radiance estimate (PPM)
  - Stochastically sample a point along eye ray (SPPM)



SPPM style: stochastic sampling along a ray

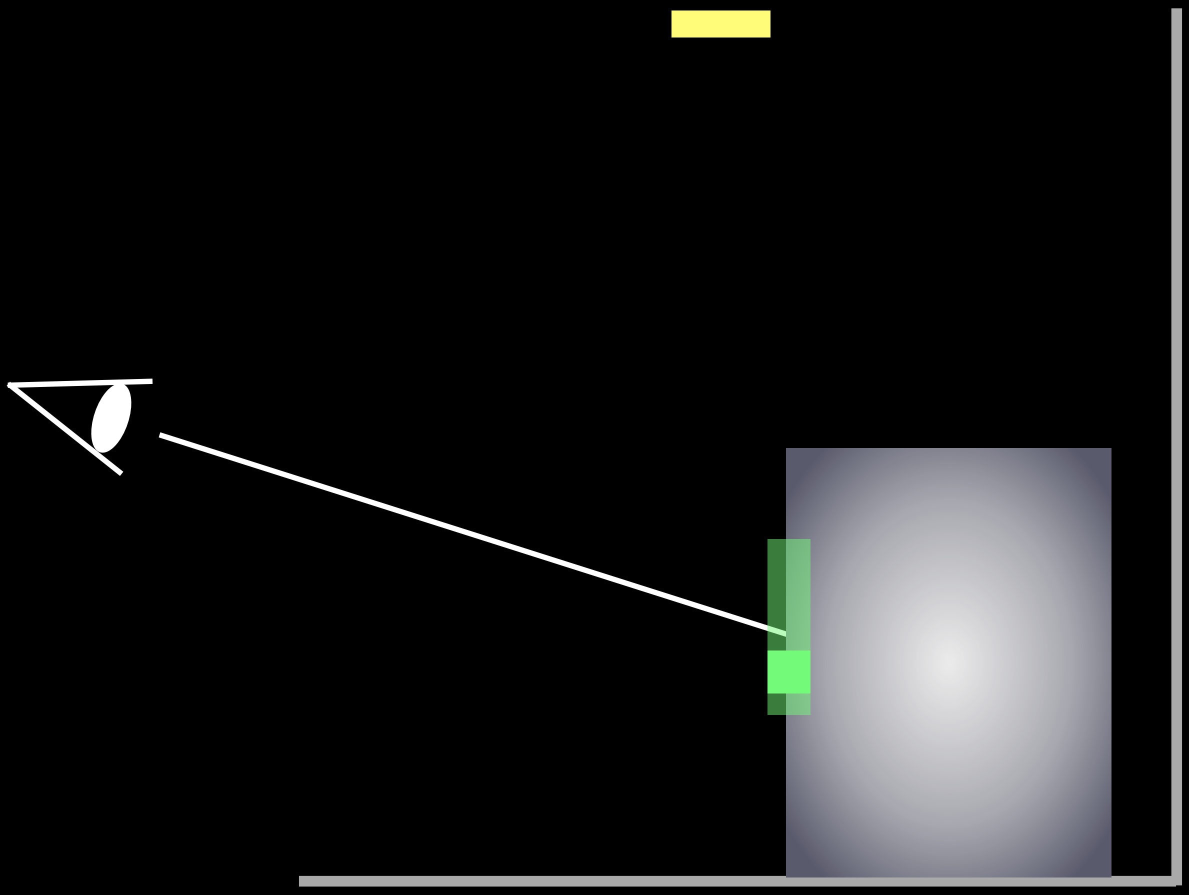
BSSRDF with PPM

- Stochastically sample a disc around the original location
  - e.g., BSSRDF Importance Sampling [King et al. 2013]



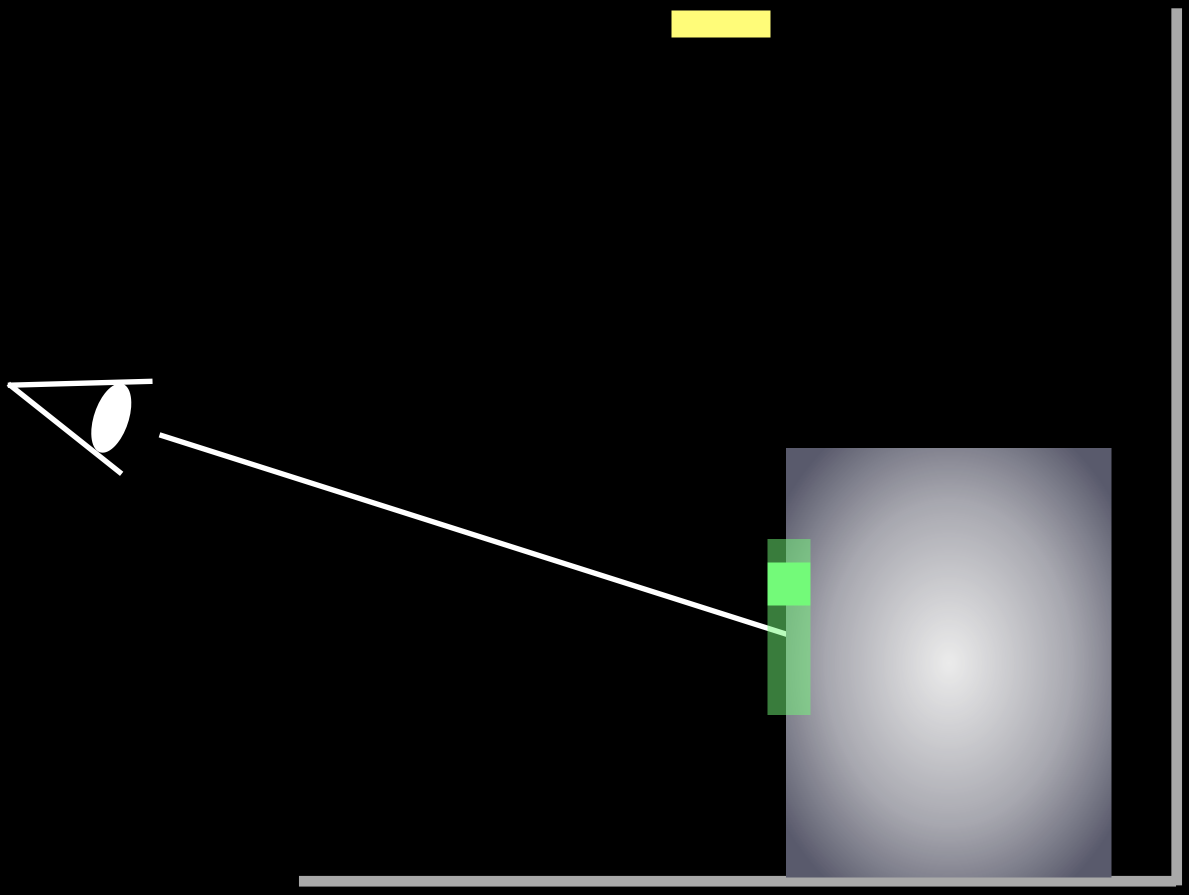
BSSRDF with PPM

- Stochastically sample a disc around the original location
  - e.g., BSSRDF Importance Sampling [King et al. 2013]



BSSRDF with PPM

- Stochastically sample a disc around the original location
  - e.g., BSSRDF Importance Sampling [King et al. 2013]

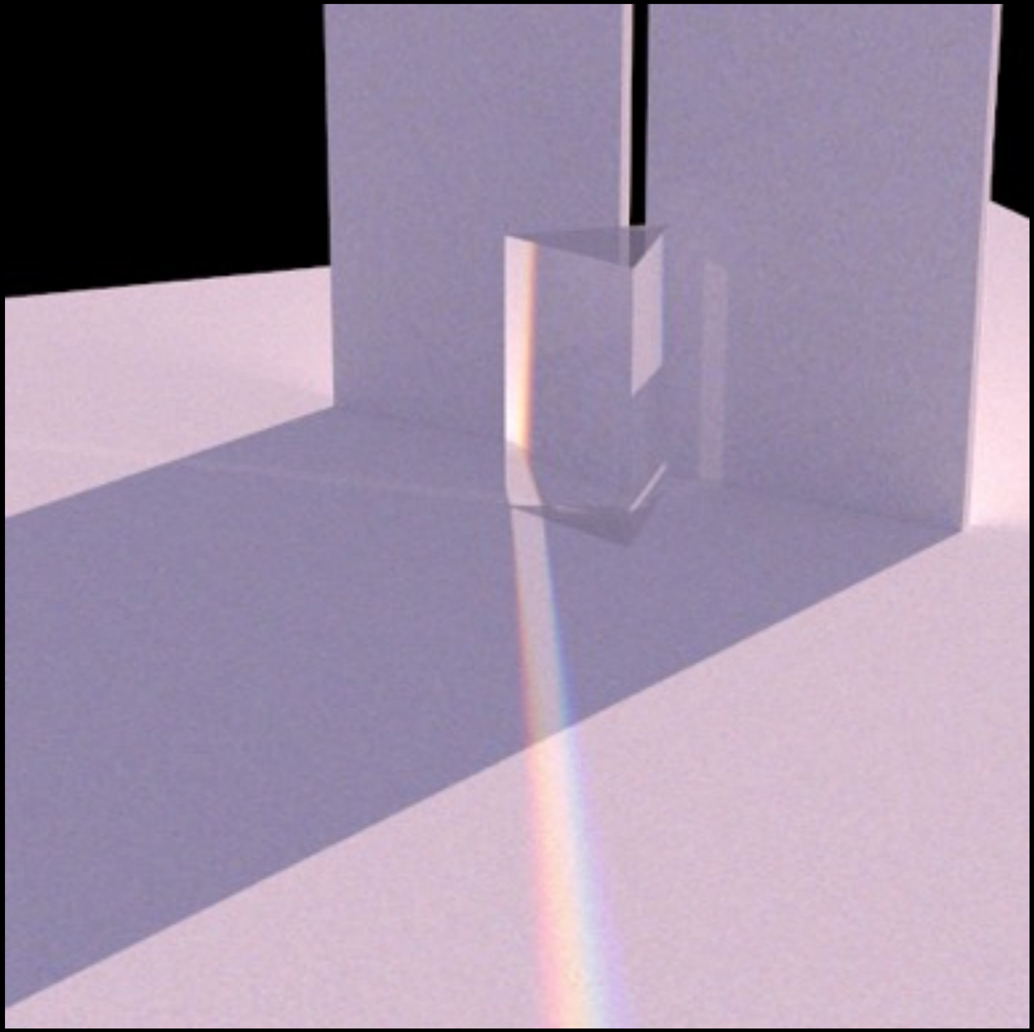


BSSRDF with PPM

- Stochastically sample a disc around the original location
  - e.g., BSSRDF Importance Sampling [King et al. 2013]

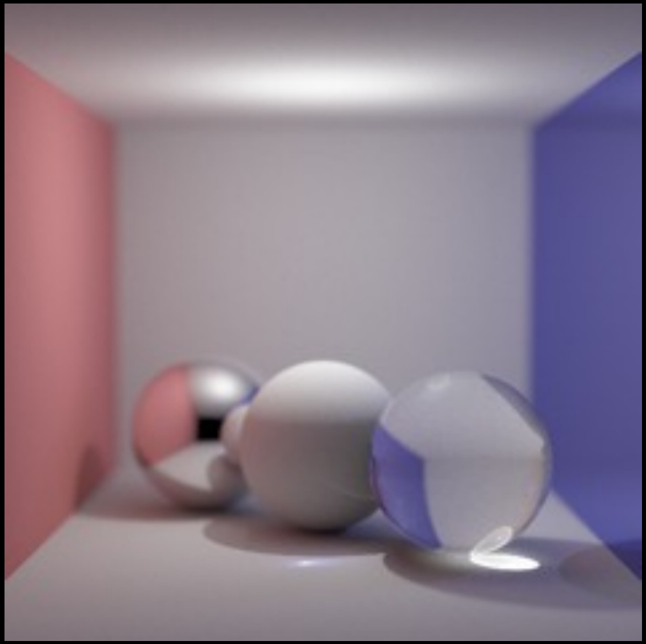
Full Spectrum Rendering

- Pick one random wavelength per iteration



GPUSPPM

- SPPM implementation using GLSL
  - Based on smallppm
  - DOF, motion blur, glossy reflection, full spectrum
  - Stochastic hashing for accel. data structure
    "Parallel Progressive Photon Mapping on GPUs", 2010



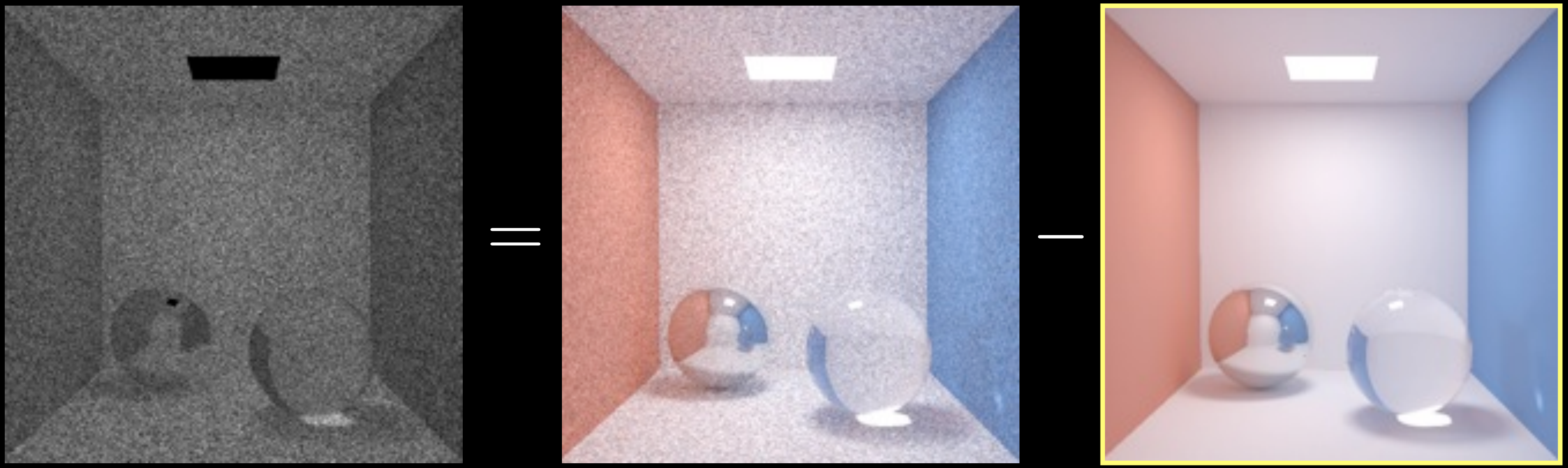
cs.au.dk/~toshiya/gpusppm.zip

How much computation is enough?

Definition of Error

- Difference between computed and exact

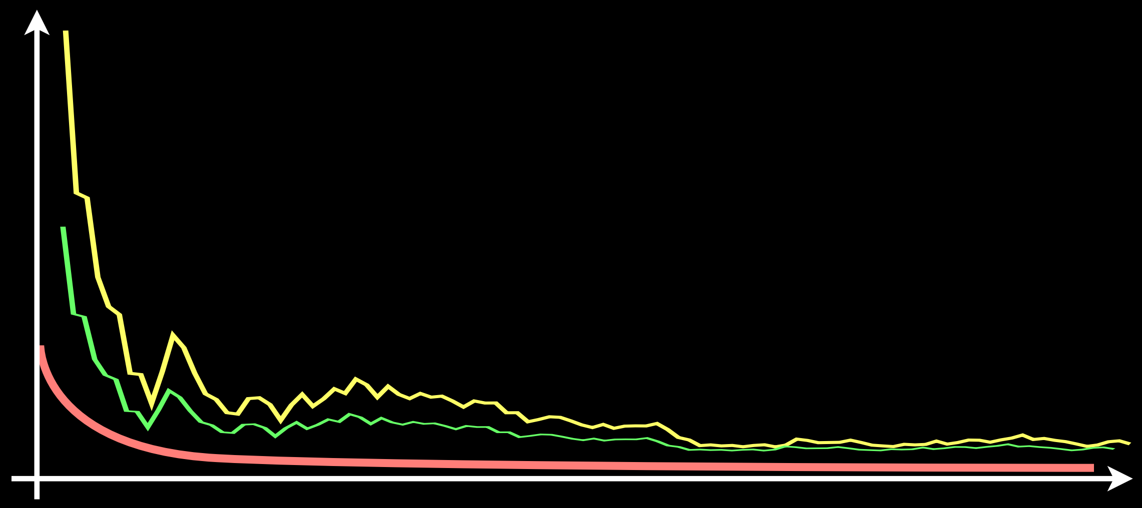
$$E_i = L_i - \boxed{L}$$ Unknown



Decomposition of Error

- Bias-Noise decomposition

$$E_i = L_i - L = B_i + N_i$$



Stochastic Error Bound Derivation

$$E_i = L_i - L = B_i + N_i$$

User-defined Probability

Stochastic error bound

$$\boxed{P(|E_i| \leq E_{b,i})} \leq \boxed{1 - \beta}$$

$$E_{b,i} = C_{i,1-\frac{\beta}{2}} \sqrt{\frac{\text{Variance}}{i}} + |B_i|$$

Stochastic Error Bound Derivation

$$E_i = L_i - L = B_i + N_i$$

$$P(|E_i| \leq E_{b,i}) \leq 1 - \beta$$

$$E_{b,i} = \boxed{C_{i,1-\frac{\beta}{2}} \sqrt{\frac{\text{Variance}}{i}}} + |B_i|$$

Error due to Noise

Stochastic Error Bound Derivation

$$E_i = L_i - L = B_i + N_i$$

$$P(|E_i| \leq E_{b,i}) \leq 1 - \beta$$

$$E_{b,i} = C_{i,1-\frac{\beta}{2}} \sqrt{\frac{\text{Variance}}{i}} + \boxed{|B_i|}$$

Error due to Bias

Bias Estimation

- Bias can be approximated by Laplacian
- Progressive estimation of Laplacian (and any derivatives)

$$B_i \approx k_2 R_i^2 \Delta L$$

k_2 constant

R_i search radius

ΔL Laplacian of radiance

Kernel

$$L_i(x) = \frac{\sum \boxed{K(x_p - x)} f_r(x, \omega, \omega_p) \Phi(x_p, x)}{\pi R_i^2}$$

Bias Estimation

- Bias can be approximated by Laplacian
- Progressive estimation of Laplacian (and any derivatives)

$$B_i \approx k_2 R_i^2 \Delta L$$

k_2 constant

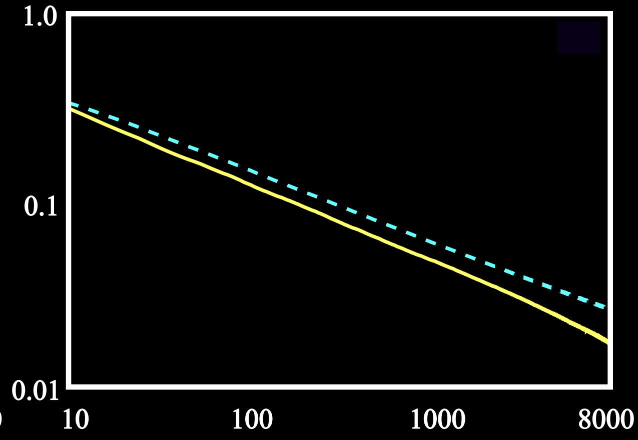
R_i search radius

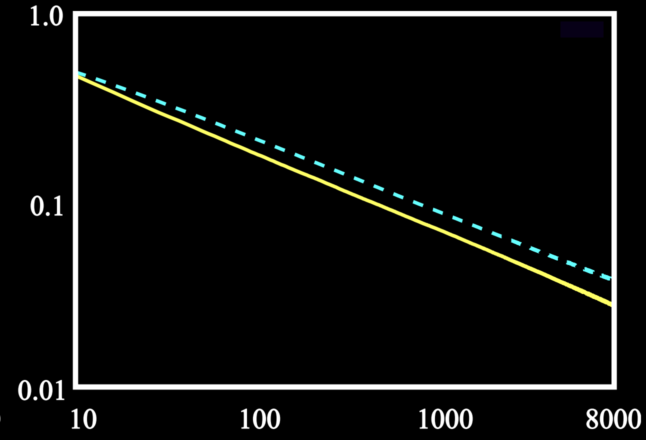
ΔL Laplacian of radiance

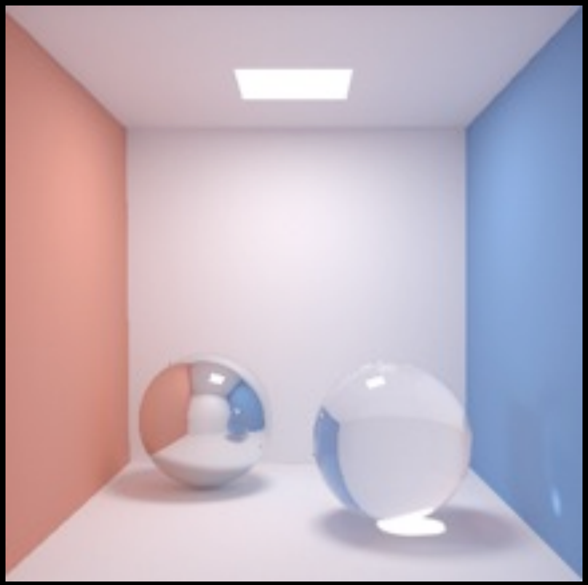
Laplacian of the kernel

$$\Delta L_i(x) = \frac{\sum \boxed{\Delta K(x_p - x)} f_r(x, \omega, \omega_p) \Phi(x_p, x)}{\pi R_i^2}$$

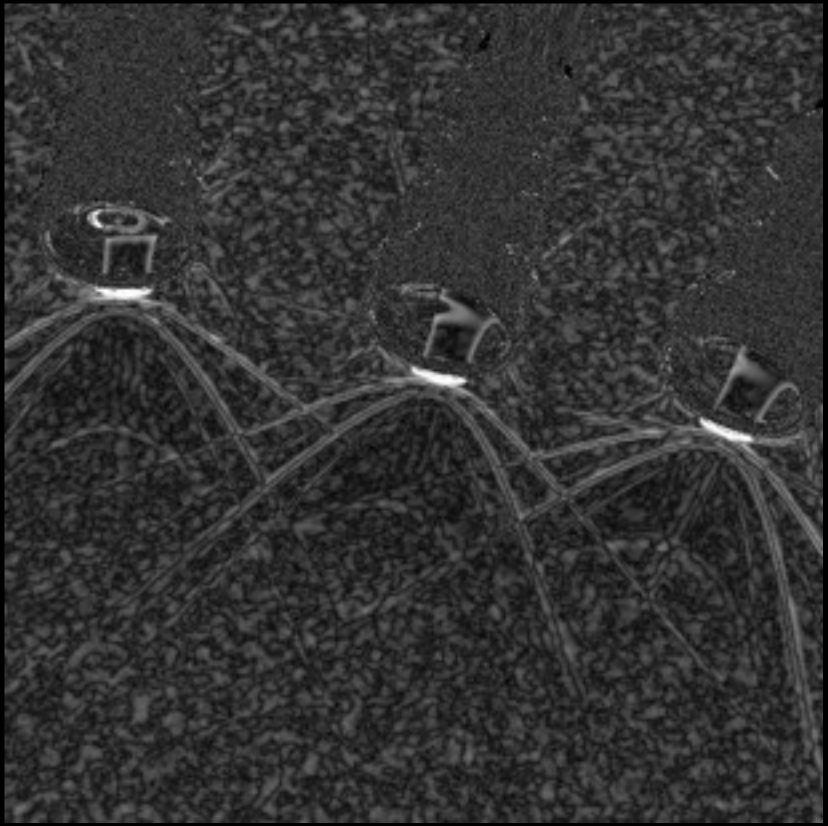
Error Estimation



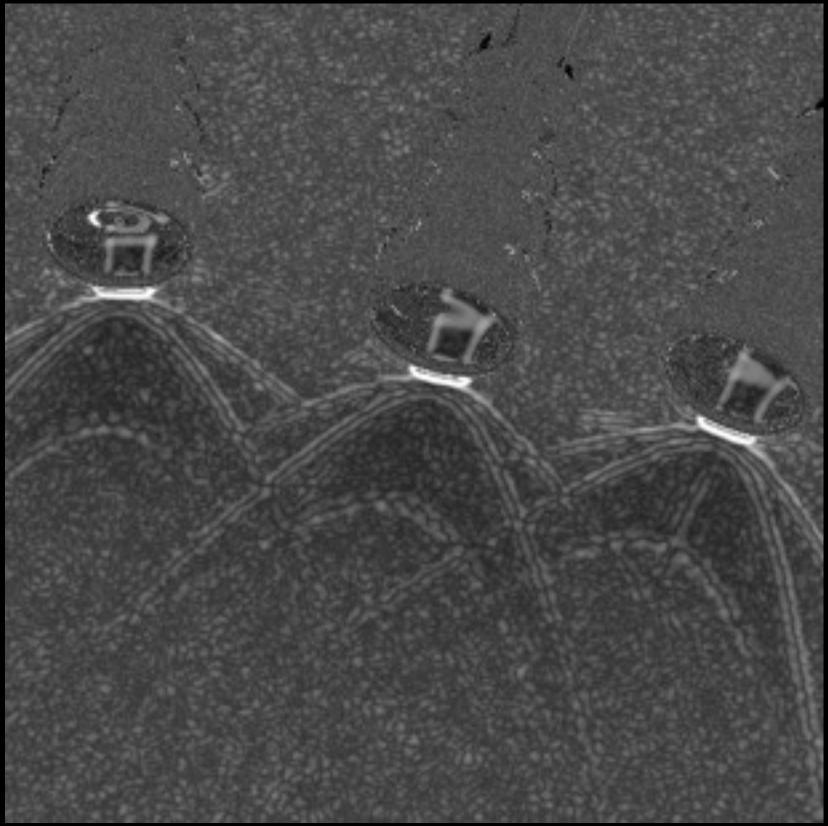




Error Estimation

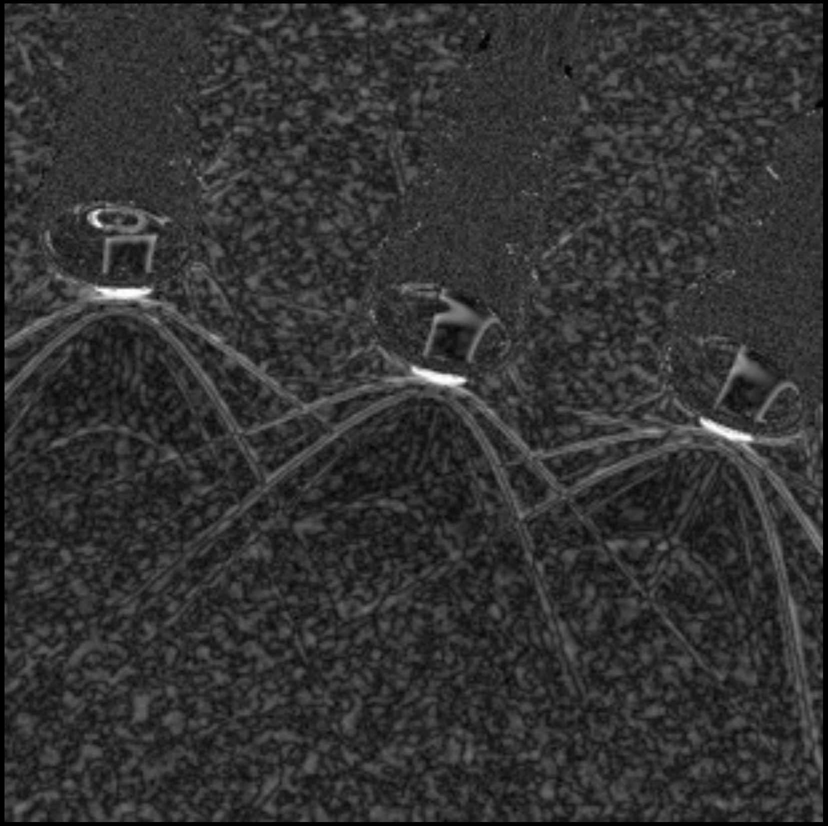


Actual Error

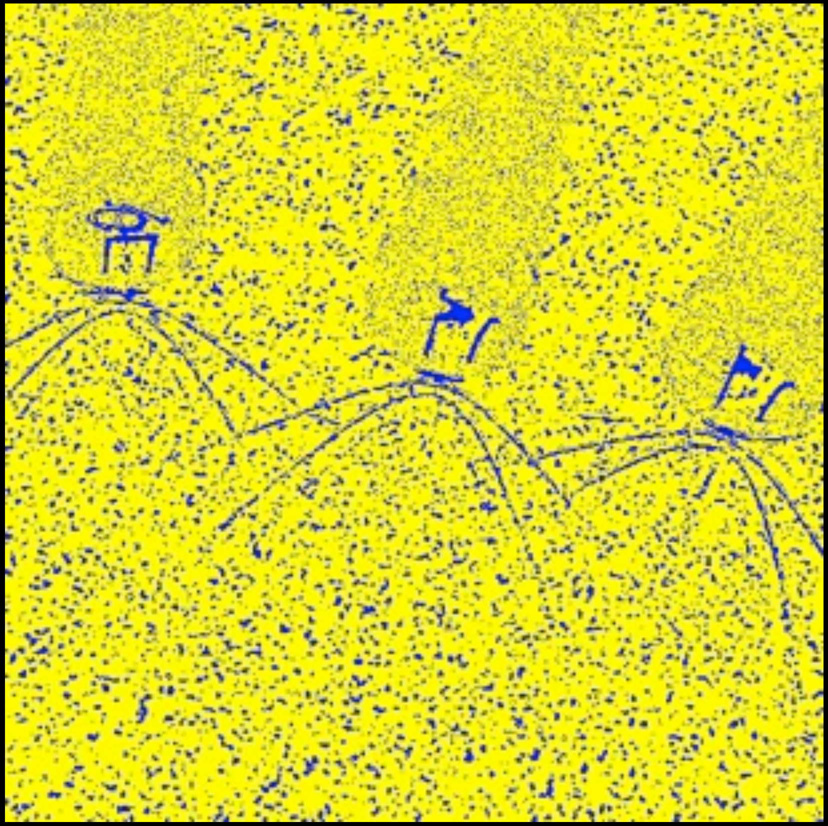


Estimated Error Bound

Error Estimation



Actual Error

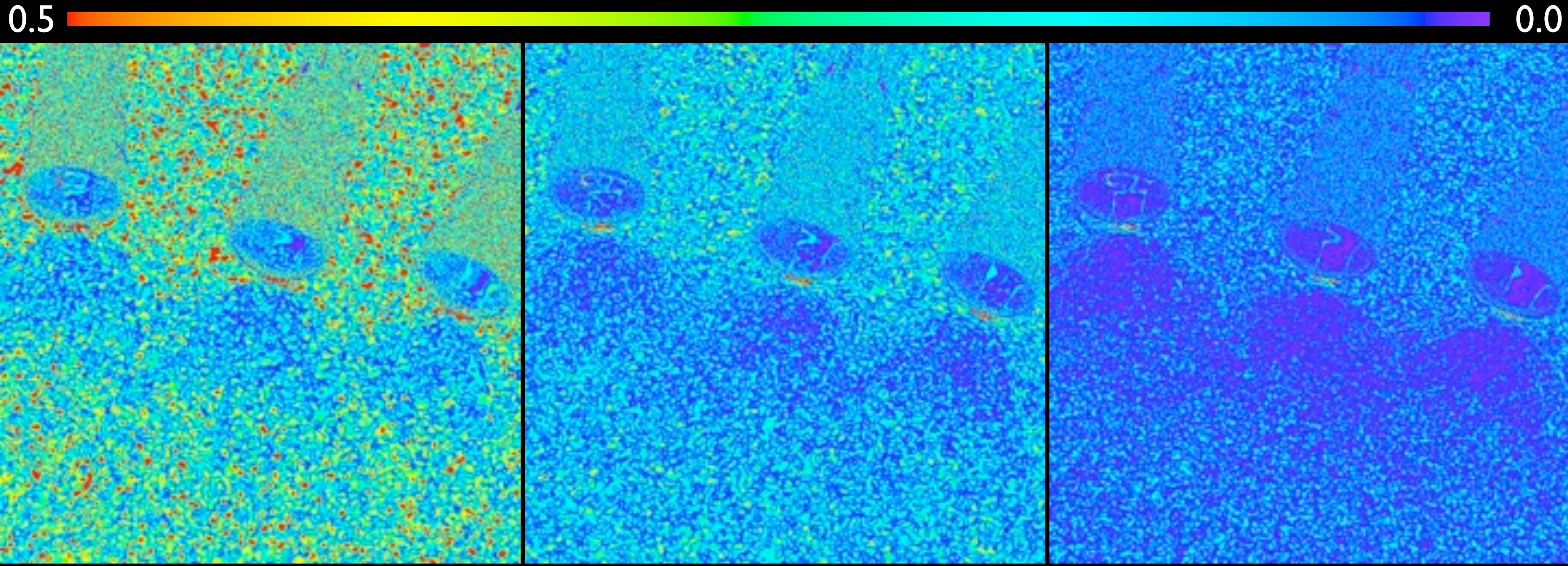


Bounded/Not bounded

Specified: 90%   Actual: 85%

Automatic Rendering Termination

SIGGRAPHASIA2013



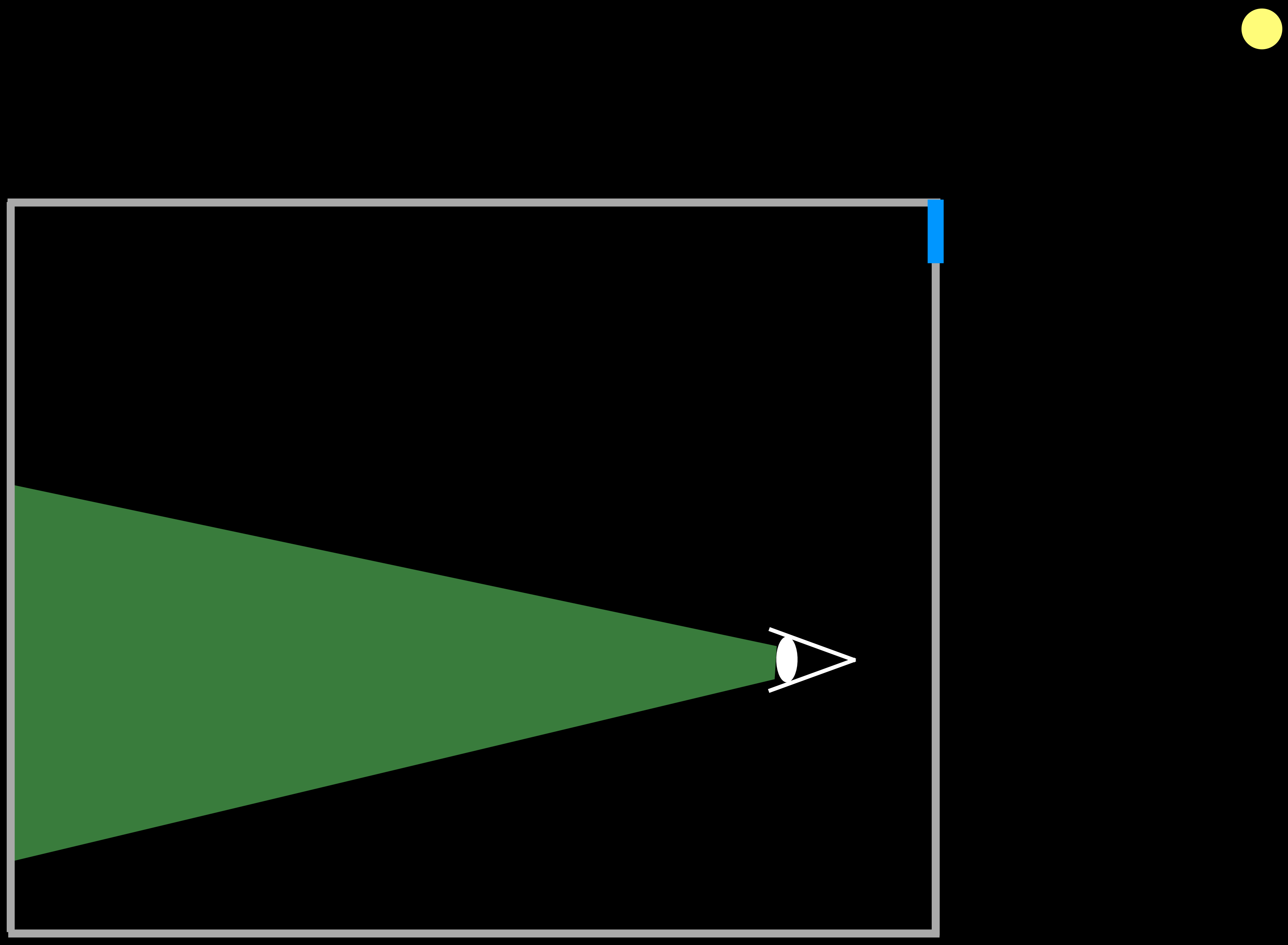
specified: 0.25
actual: 0.1916

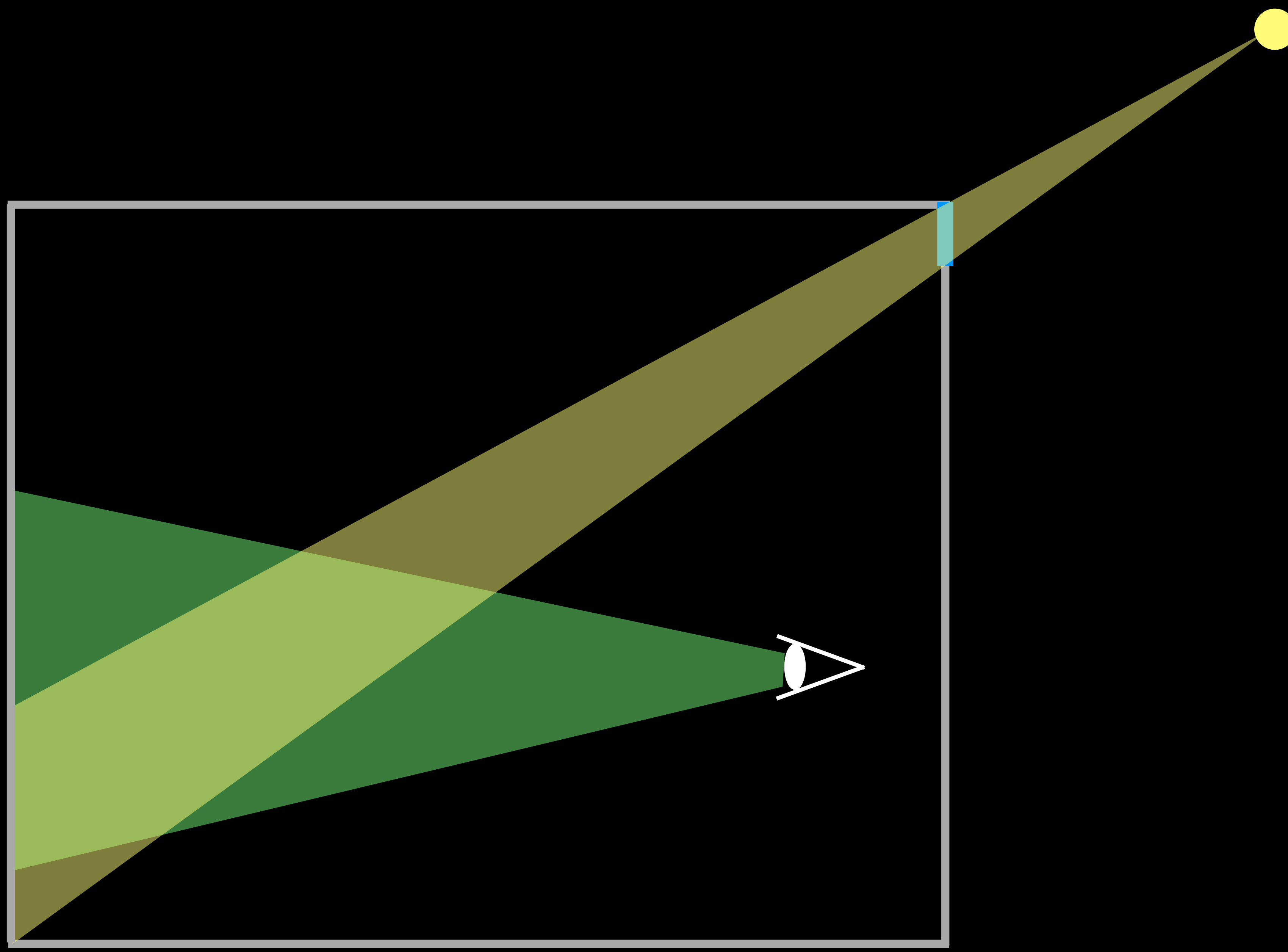
specified: 0.125
actual: 0.09294

specified: 0.0625
actual: 0.04482

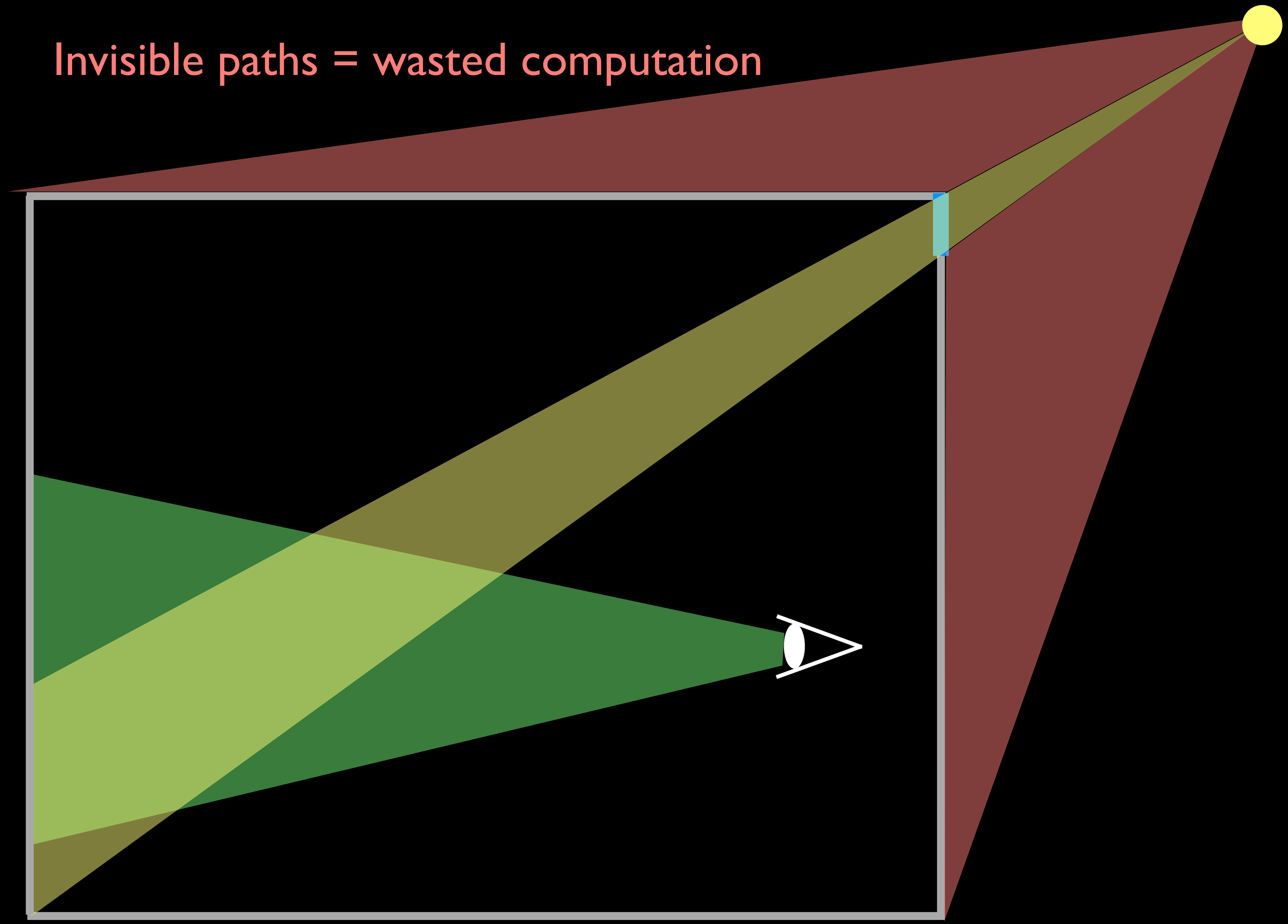
1.3 times overestimation on average


Light source
Window
Eye





Invisible paths = wasted computation



Metropolis Light Transport

Ideal

- Can we combine these two algorithms?
  - MLT: Efficient for difficult lighting scenarios
  - PPM: Robust to complex types of light paths

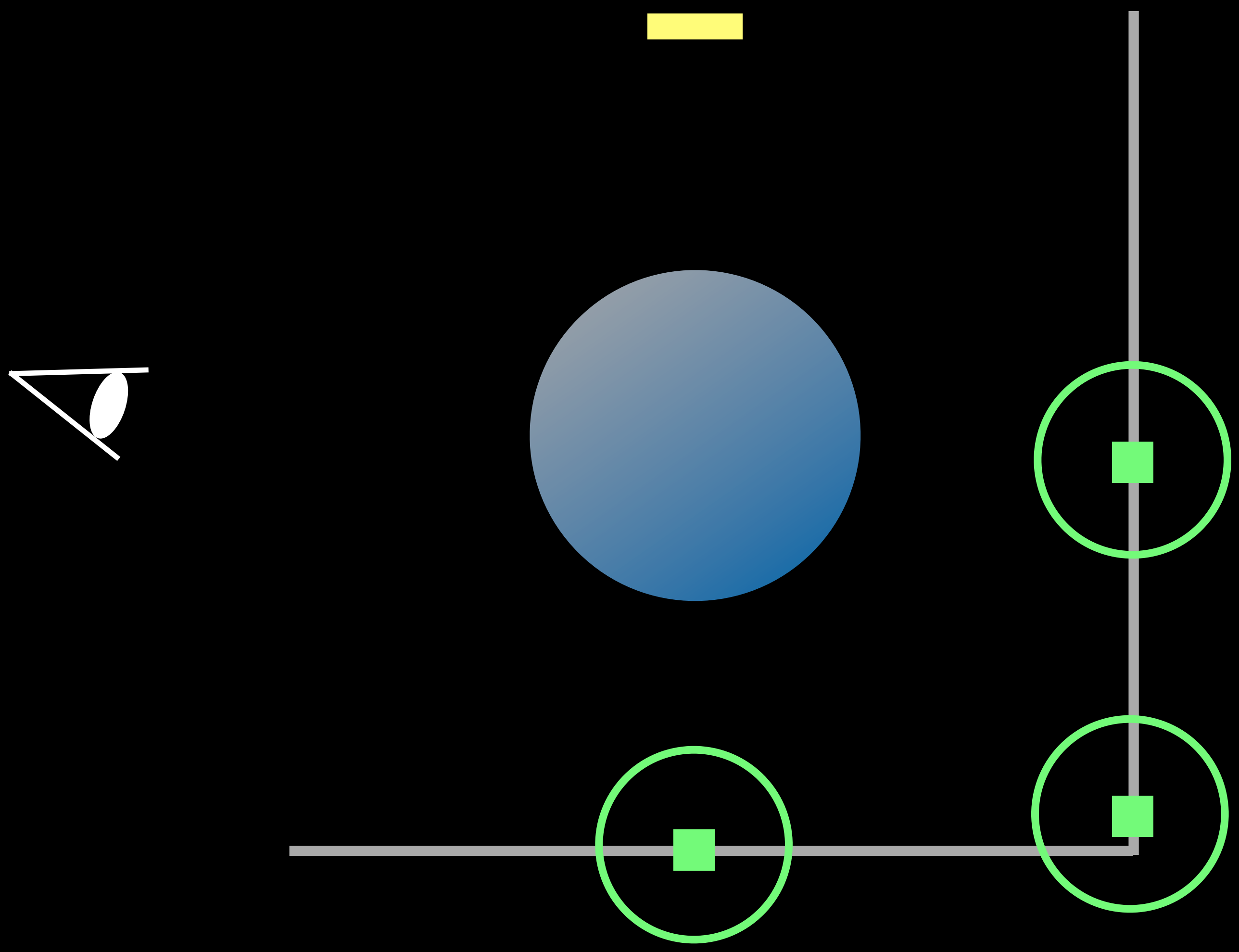
Ideal

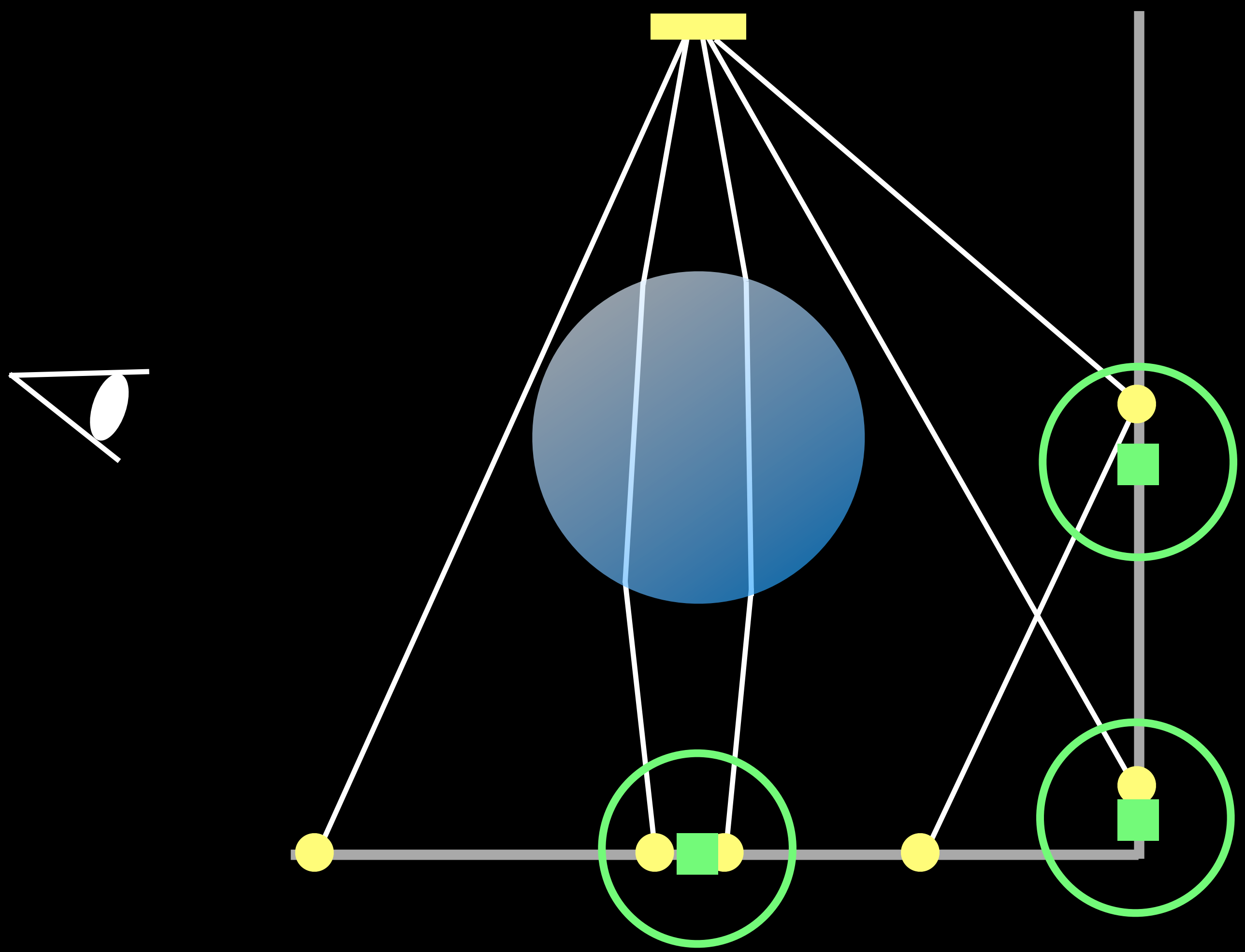
- Can we combine these two algorithms?
  - MLT: Efficient for difficult lighting scenarios
  - PPM: Robust to complex types of light paths

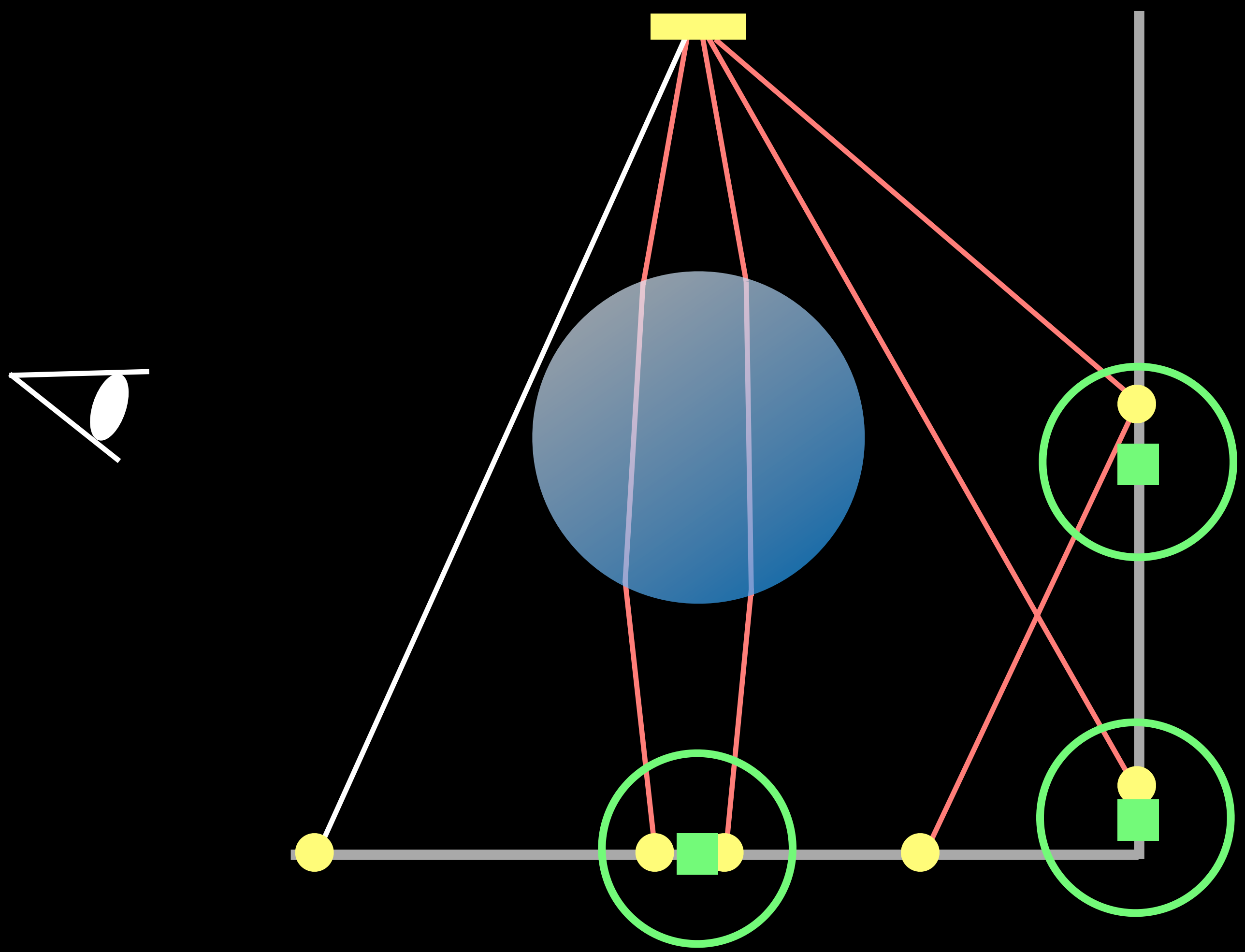
MLT + PPM

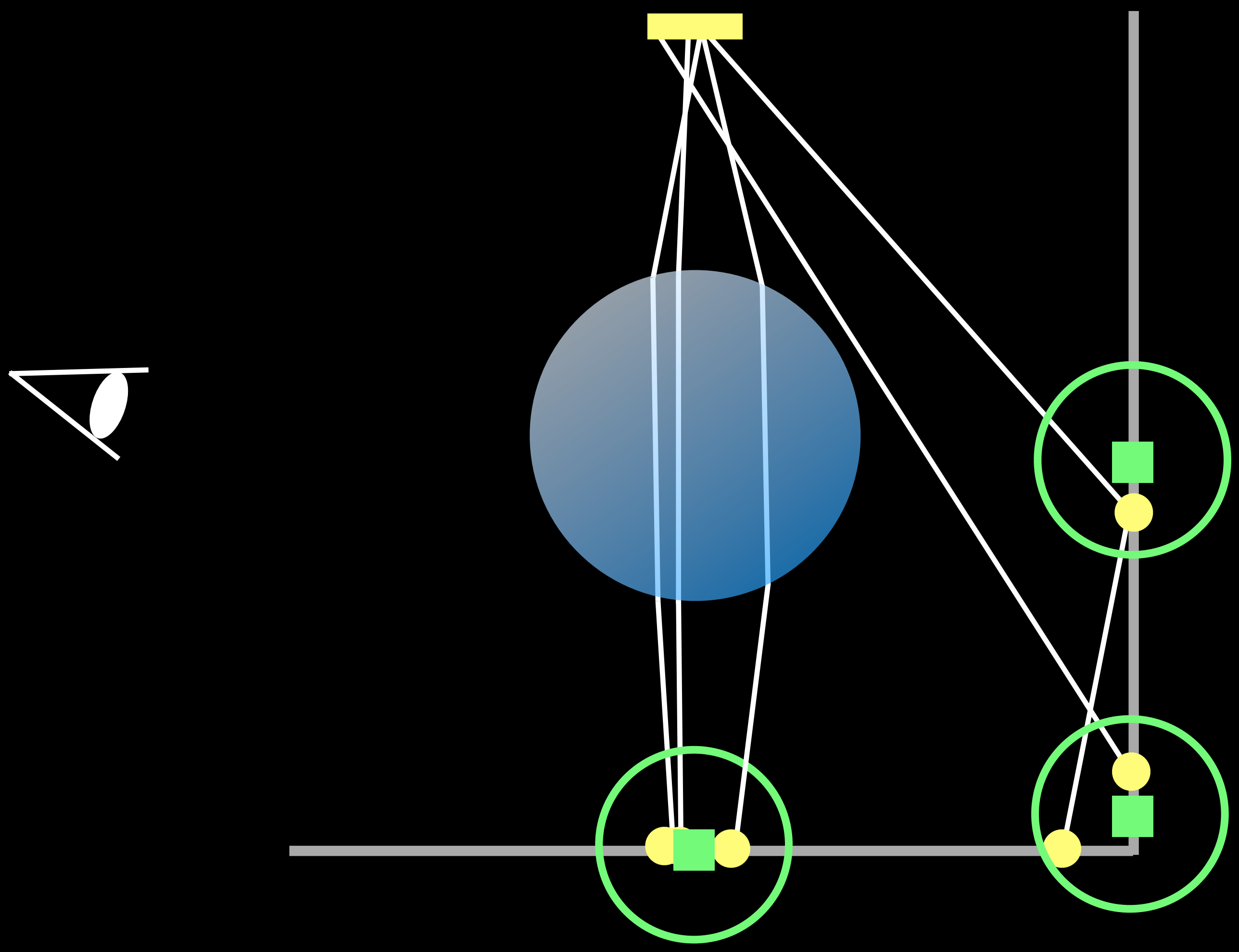
Key Observation

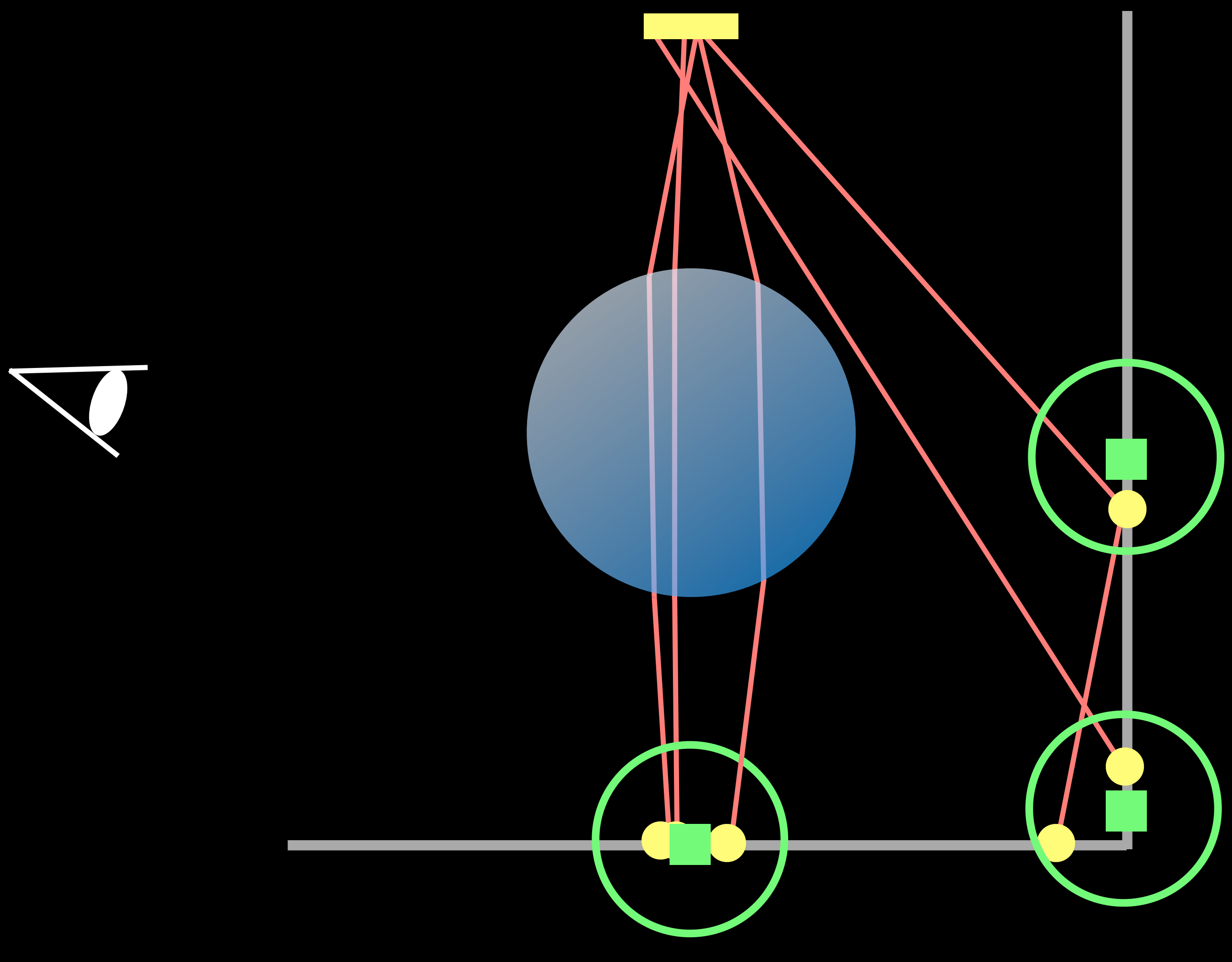
- We can determine whether a photon path is visible or not
  - Because PPM stores visible points from the eye
  - Contributed to at least one visible point = visible





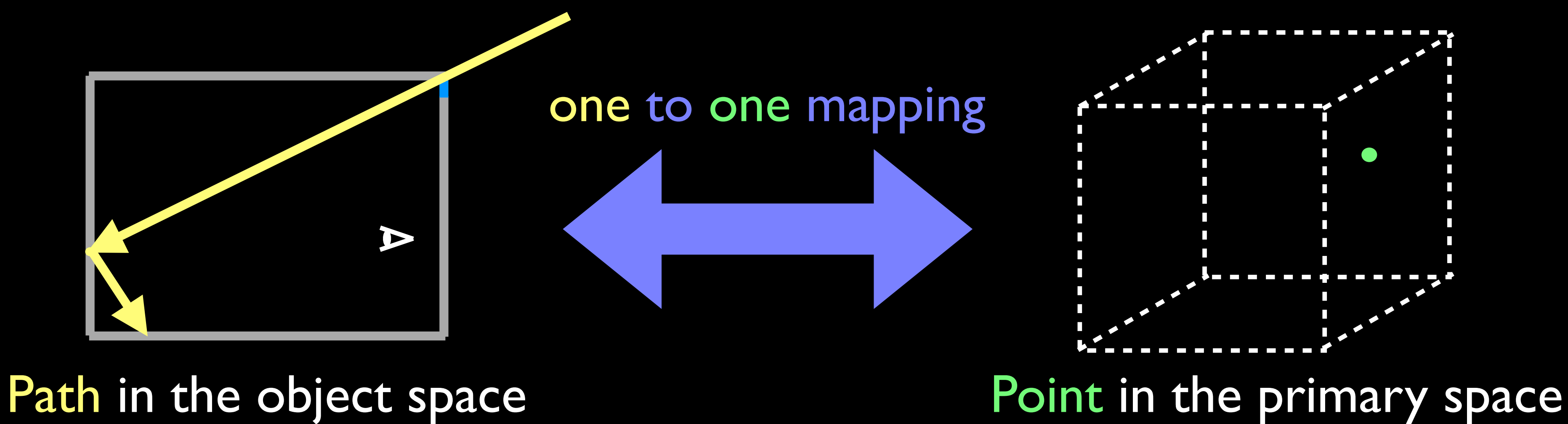






Primary Space

- Mapping a path to a point [Kelemen et al. 2002]
- Path = vector of random numbers $\vec{u} = (\xi_1, \cdots, \xi_N) \in (0,1)^N$



Visibility-based Sampling

- Consider space of random numbers
- Photon path visibility function

If the photon is not visible:

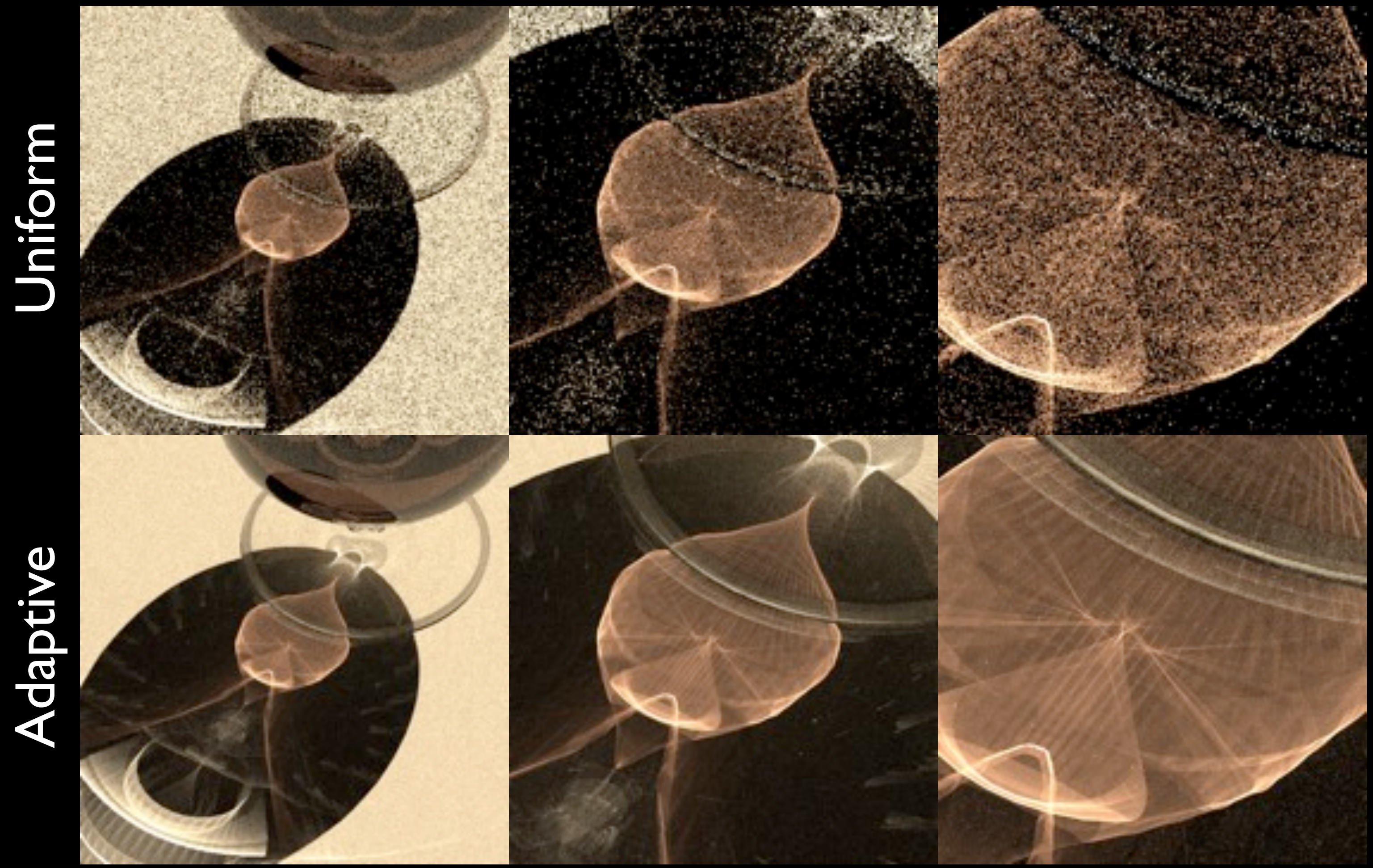
$$V\left(\vec{u}\right) = 0$$

If the photon is visible:

$$V\left(\vec{u}\right) = 1$$

Sample $V\left(\vec{u}\right)$ using Markov chain Monte Carlo Methods

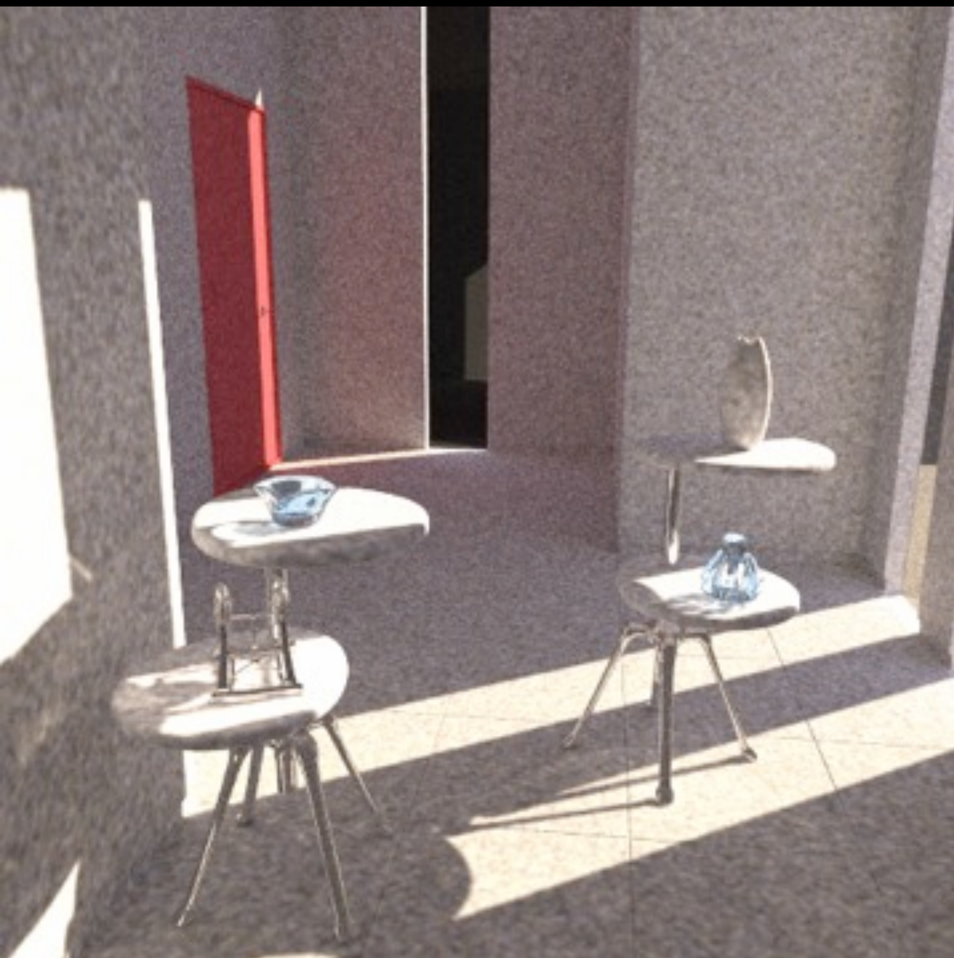
Small-scale Lighting Details



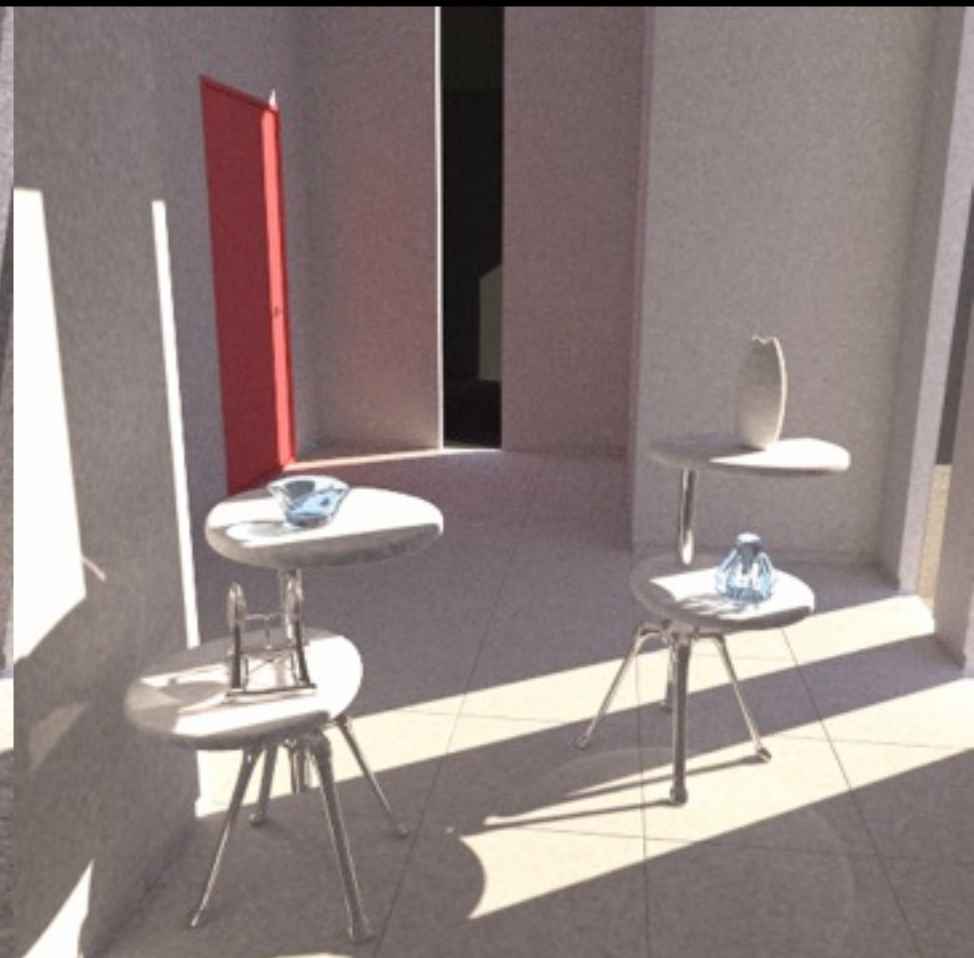
Automatic Parameter Tuning

Value is too small | Automatic | Value is too large

Sunlit Room



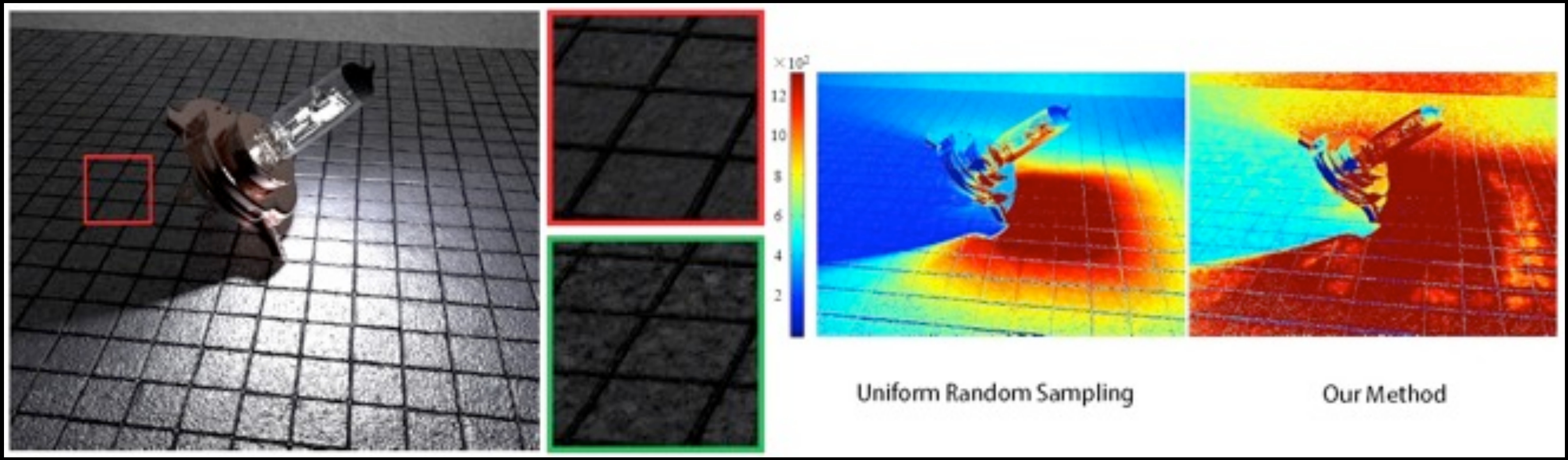
Uniform



Adaptive

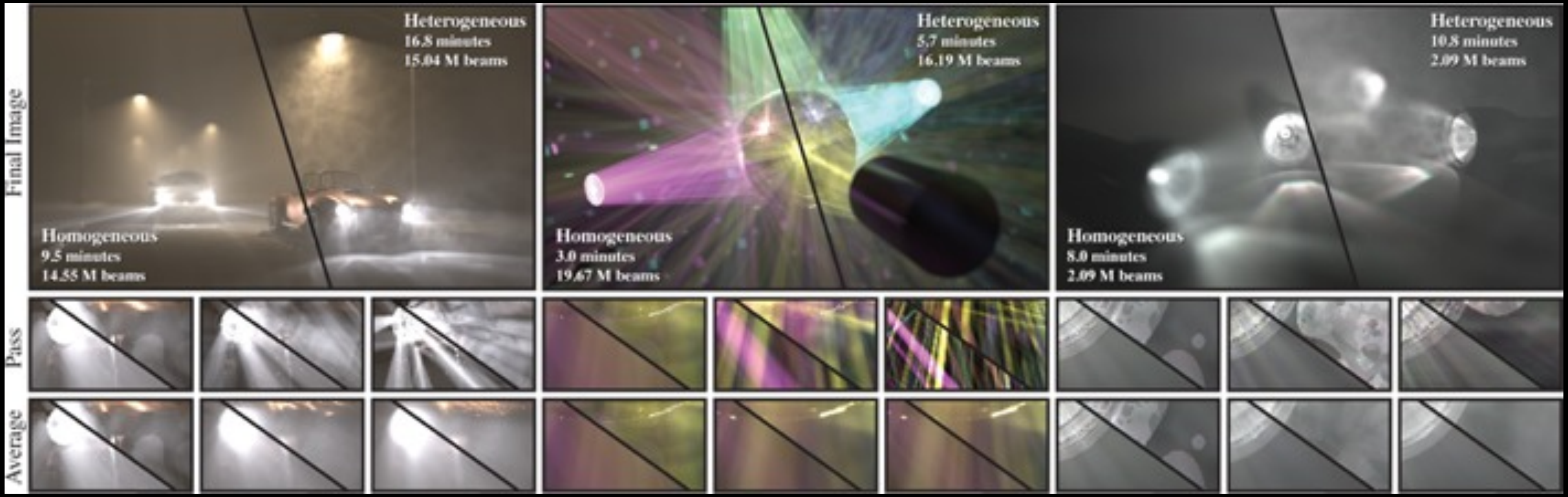
Some Related Work

- Adaptive photon tracing based on photon density on the image [Chen et al. 2011]



Some Related Work

- Progressive photon beams [Jarosz et al. 2011]
  - Wojciech will talk more about it



Some Related Work

- Efficient rendering of dynamic scenes [Weiss and Grosch 2012]

Some Related Work

- Combine density estimation and MC integration [Hachisuka et al. 2012, Georgiev et al. 2012]
  - Iliyan will talk more about it

FAQs

- Q: Is PPM unbiased?

FAQs

- Q: Is PPM unbiased?
- A: It is biased and consistent, but does not matter in practice.

$$\mathrm{E}[X] = \lim_{\boxed{N \to \infty}} \sum_{i=1}^{N} x_i$$

BOTH unbiased and consistent methods need inf. samples!

FAQs

- Q: Is PPM unbiased?
- A: It is biased and consistent, but does not matter in practice.

$$\mathrm{E}[X] = \lim_{\boxed{N \to \infty}} \sum_{i=1}^{N} x_i$$

BOTH unbiased and consistent methods need inf. samples!

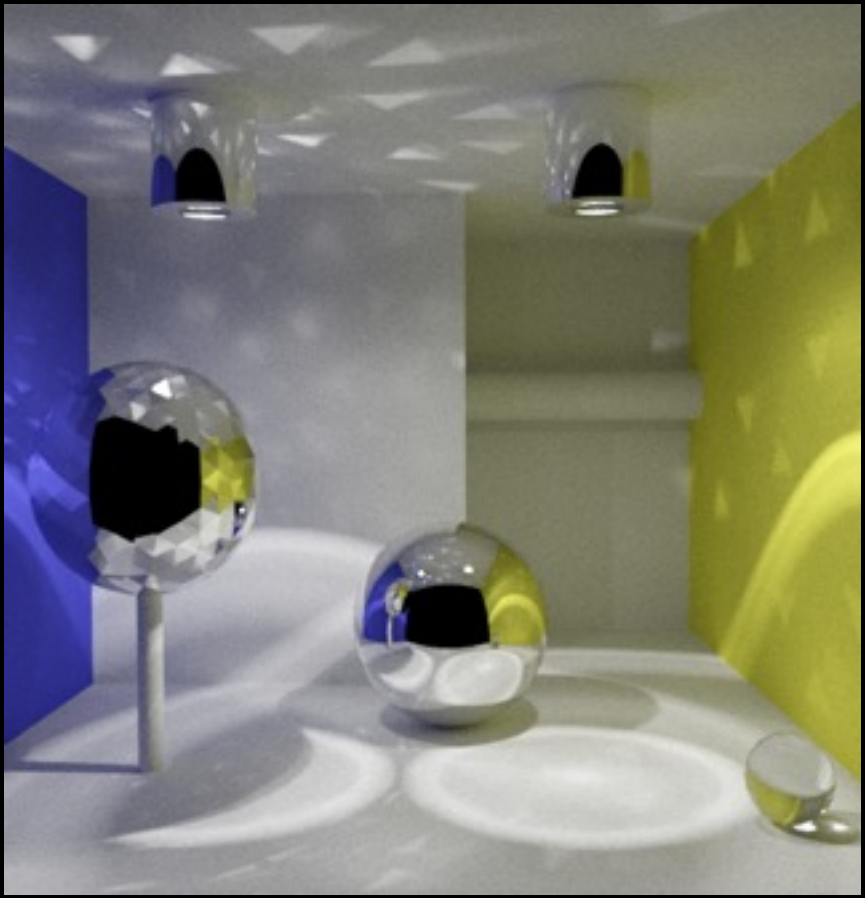
"Five Common Misconceptions about Bias in Light Transport Simulation"
cs.au.dk/~toshiya/misc.pdf

FAQs

- Q: Do we still use global + caustics separation?

FAQs

- Q: Do we still use global + caustics separation?
- A: No. Just render everything as one.

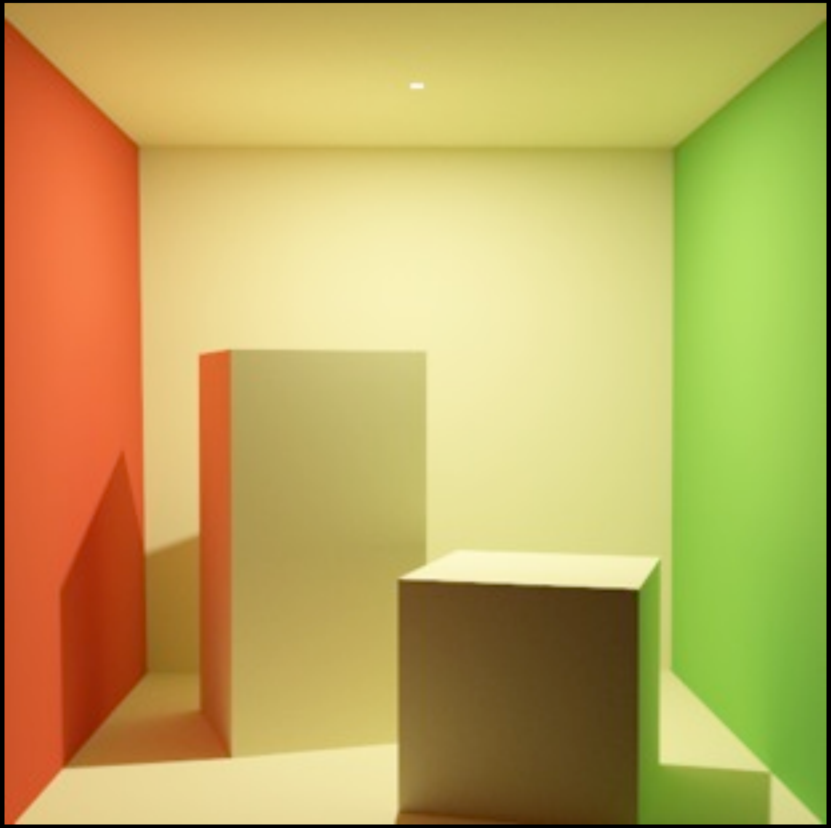


FAQs

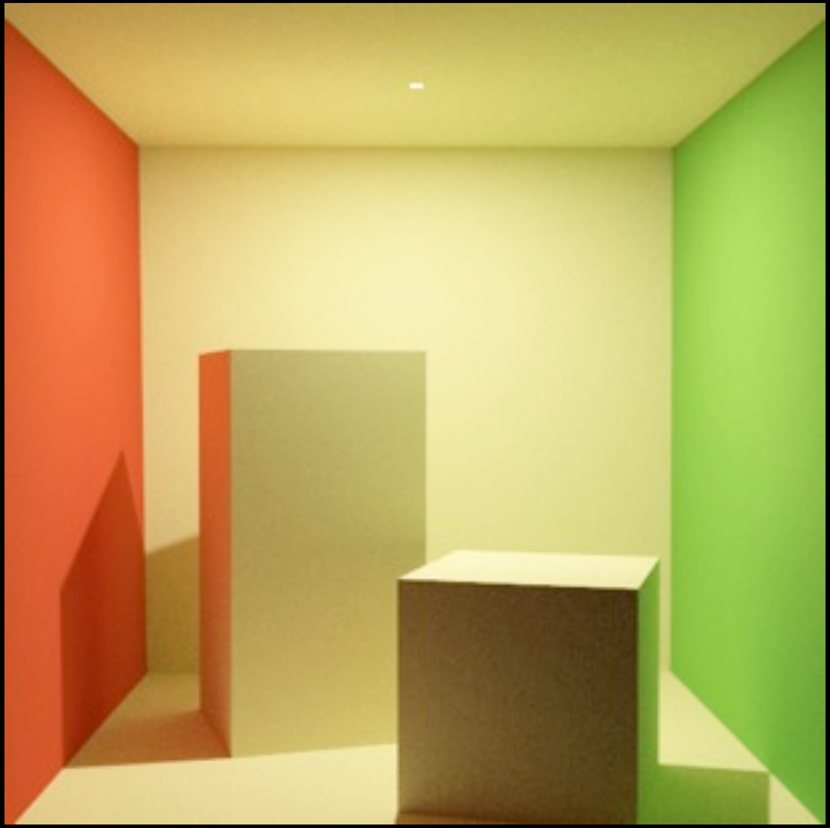
- Q: Is PPM slower for diffuse scenes than other methods?

FAQs

- Q: Is PPM slower for diffuse scenes than other methods?
- A: True, but not much, and you can do more.



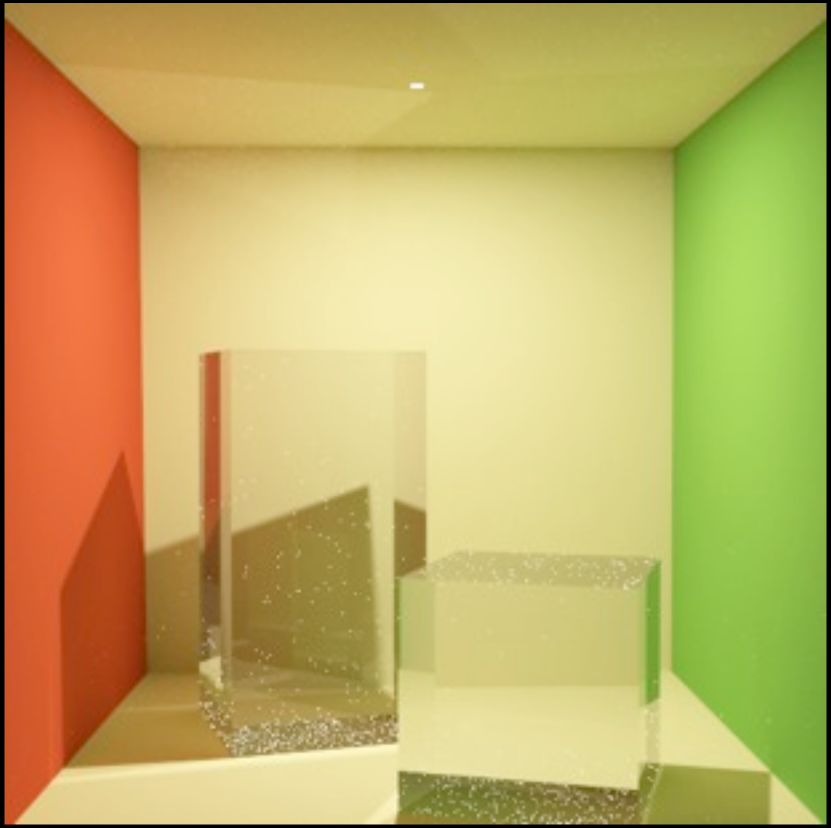
Bidirectional PT



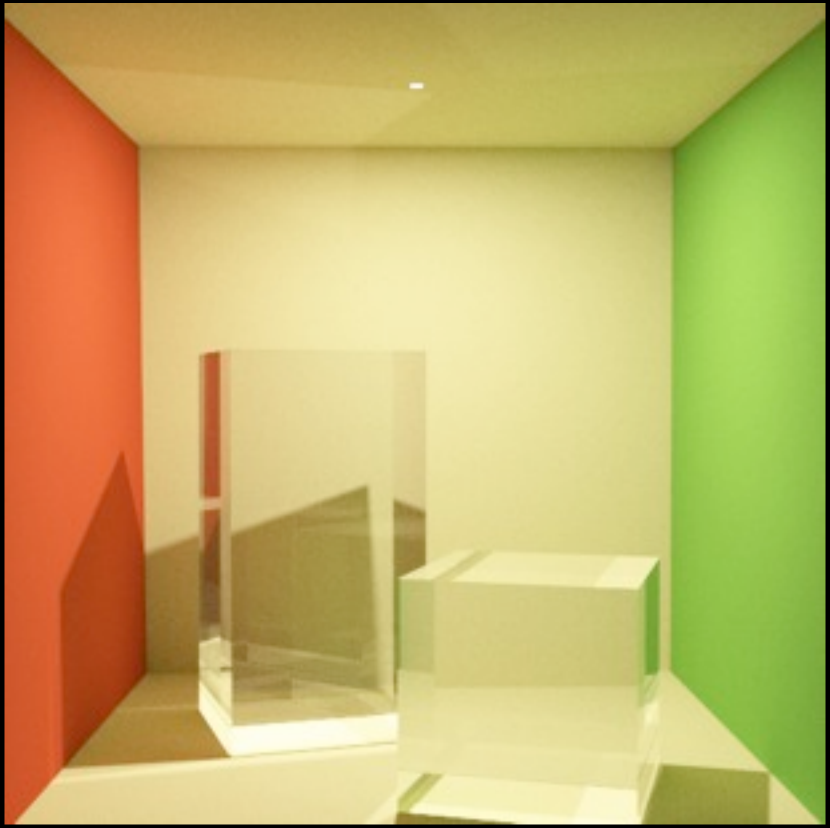
PPM

FAQs

- Q: Is PPM slower for diffuse scenes than other methods?
- A: True, but not much, and you can do more.



Bidirectional PT



PPM

Summary

- SPPM = PPM + Distributed Ray Tracing
- Error estimation is available
- Adaptive photon tracing based on visibility
- Lots of useful extensions

- My opinion: (S)PPM + extensions is very hard to "break"
  - Just works fine in really many cases

More details

- cs.au.dk/~toshiya

Next Talk

- More advanced and efficient radius reduction