Stochastic Progressive Photon Mapping

Toshiya Hachisuka Henrik Wann Jensen

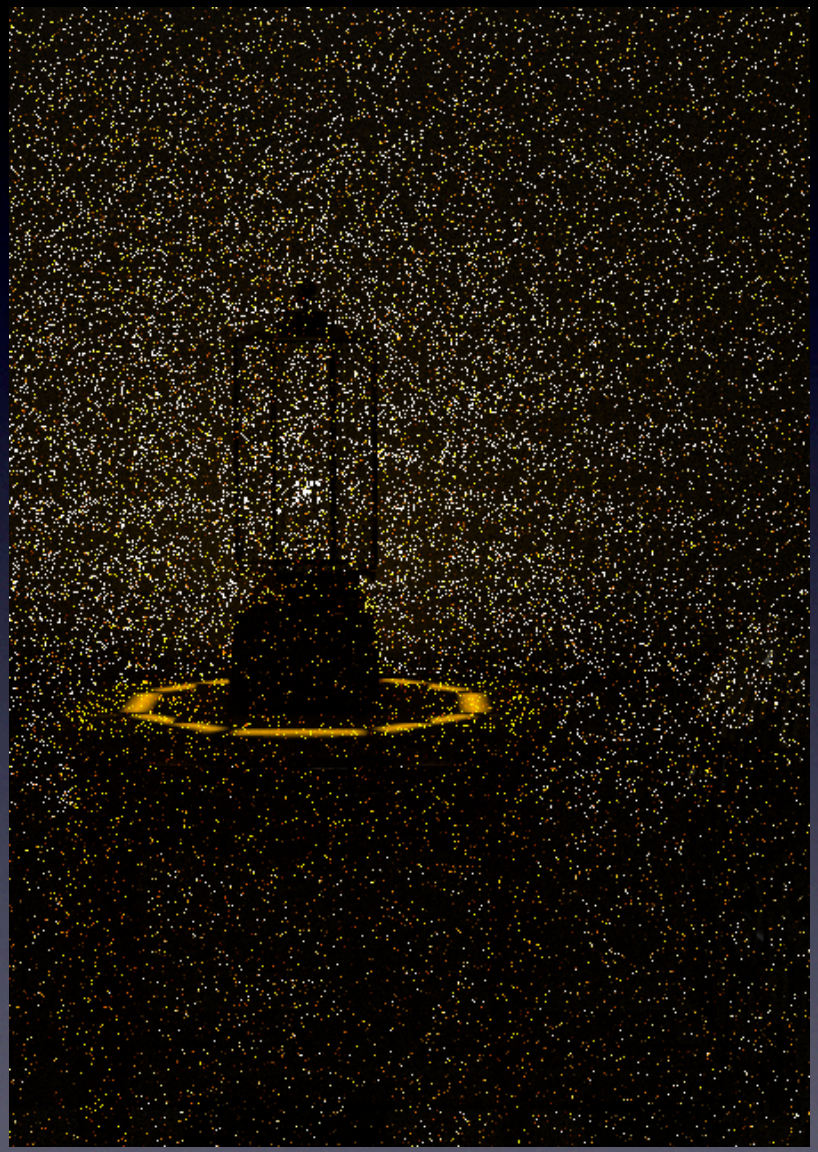
University of California, San Diego

(c)Y. Kimura

Global Illumination Algorithms

- Path Tracing [Kajiya 86]
- Light Tracing [Arvo 86][Dutré 93]
- Bidirectional Path Tracing [Lafortune 93][Veach 95]
- Photon Mapping [Jensen 95]
- Density Estimation [Shirley 95]
- Instant Radiosity [Keller 97]
- Metropolis Light Transport [Veach 97]
- Lightcuts [Walter 05]
- Energy Redistribution Path Tracing [Cline 05]
- ...

Progressive Photon Mapping [Hachisuka 08]



Path Tracing

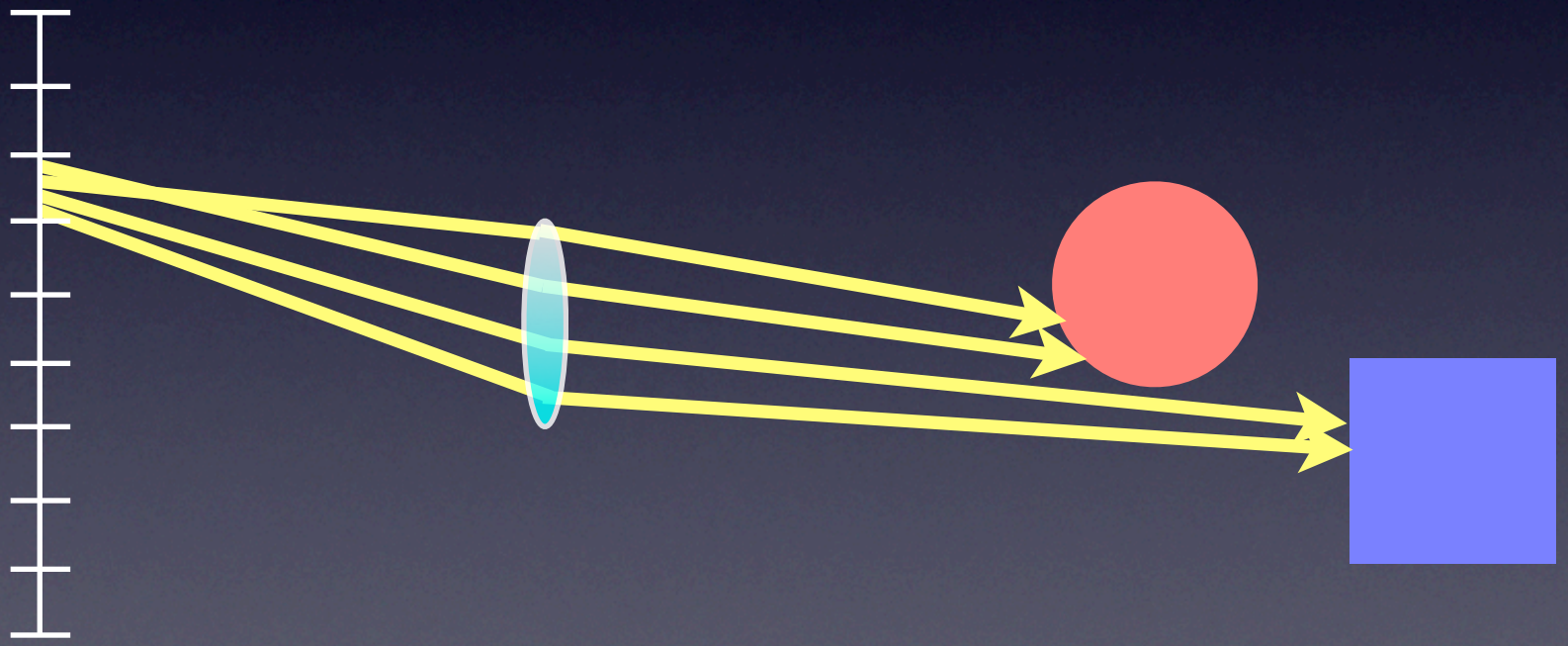
Progressive Photon Mapping

展示
デジタル一眼レフカメラ D3
交換レンズ AF-S NIKKOR 14-24mm F2.8G
カットボディ

MURRAYS
13

Distributed Ray Tracing

- Computes average radiance [Cook et al. 84]



Distributed Ray Tracing

- Computes average radiance [Cook et al. 84]
  - Depth-of-field - visible region through lens
  - Motion blur - time
  - Anti-aliasing - pixel footprint
  - Glossy reflections - hemisphere

Stochastic Progressive Photon Mapping

First robust algorithm for computing *all* types of light transport with distributed ray tracing effects

Stochastic Progressive Photon Mapping

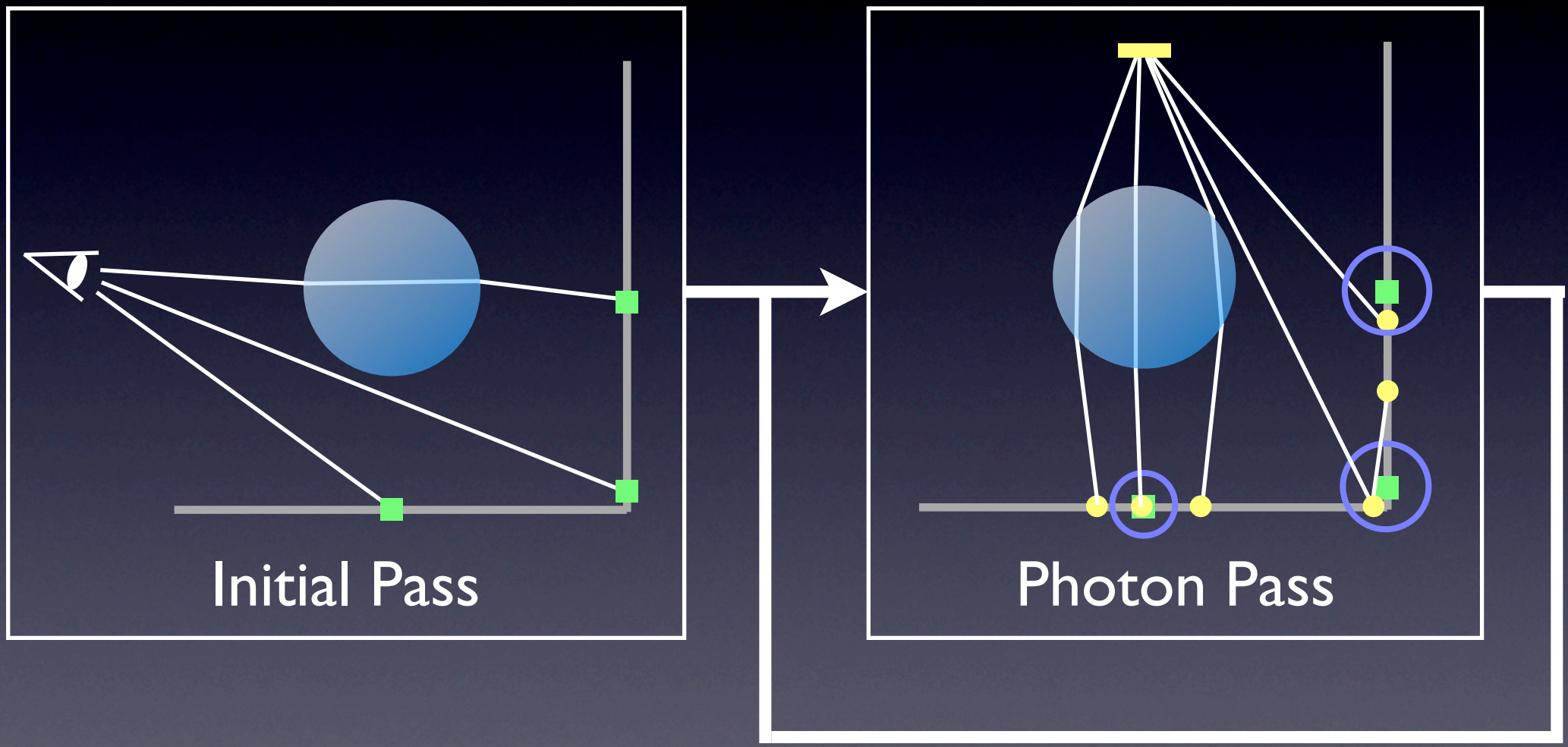
- Extension of progressive photon mapping
  - Consistent average radiance estimation
  - Can handle distributed ray tracing effects

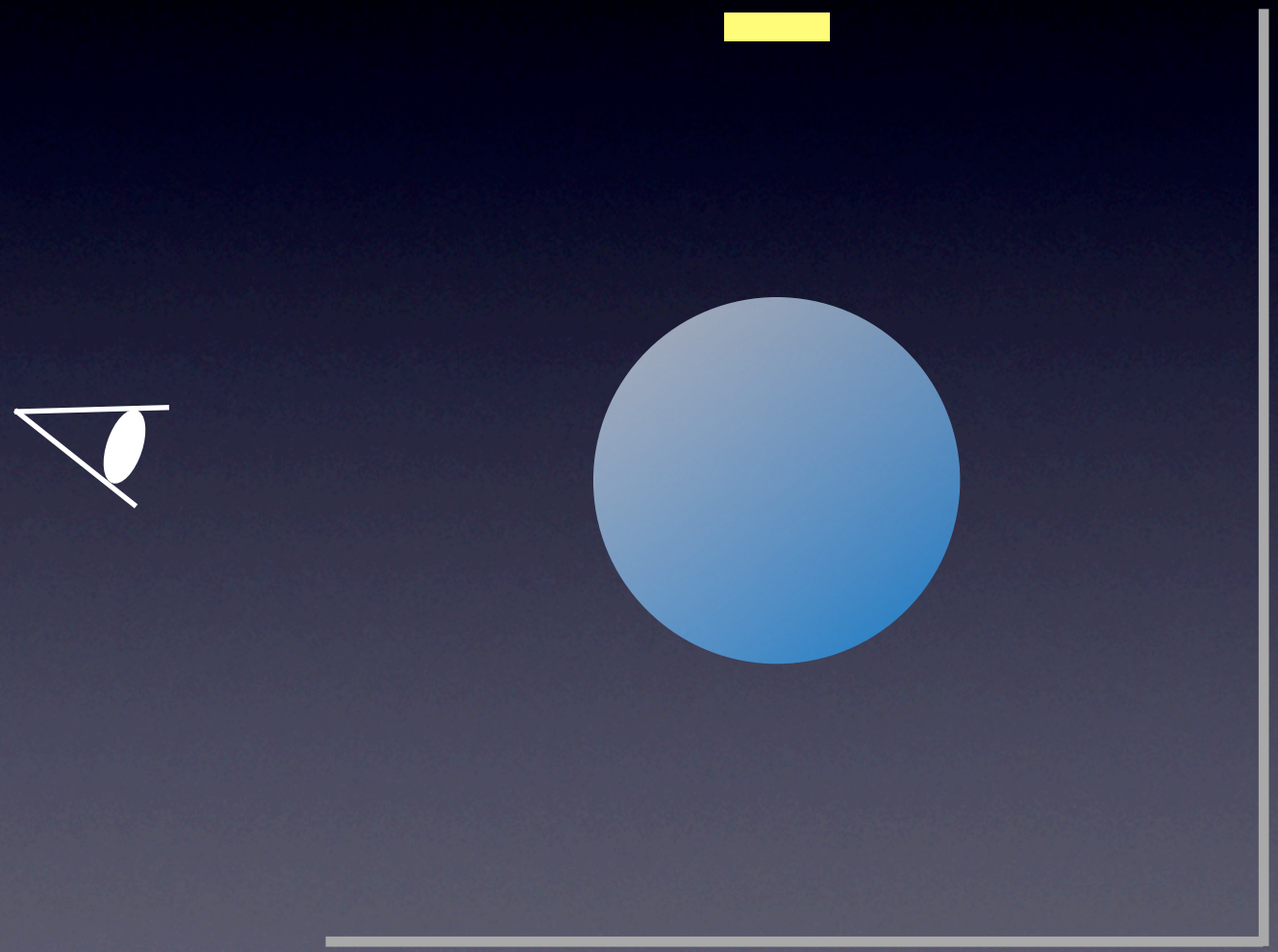
Stochastic Progressive Photon Mapping

- Extension of progressive photon mapping
  - Consistent average radiance estimation
  - Can handle distributed ray tracing effects
  - Robust for *all* light path
  - Arbitrary accuracy using finite memory
  - Easy to implement

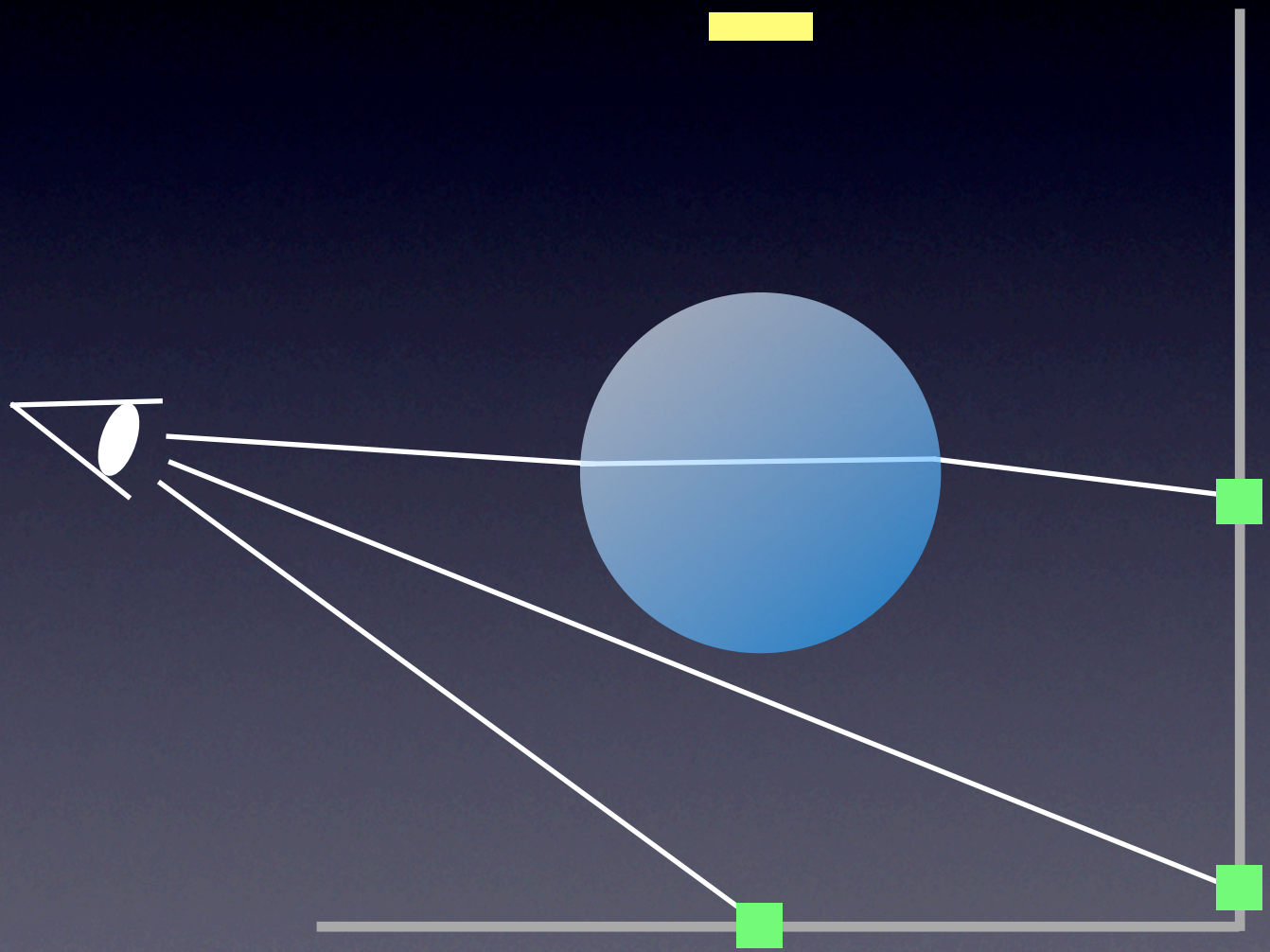
Progressive Photon Mapping

Overview

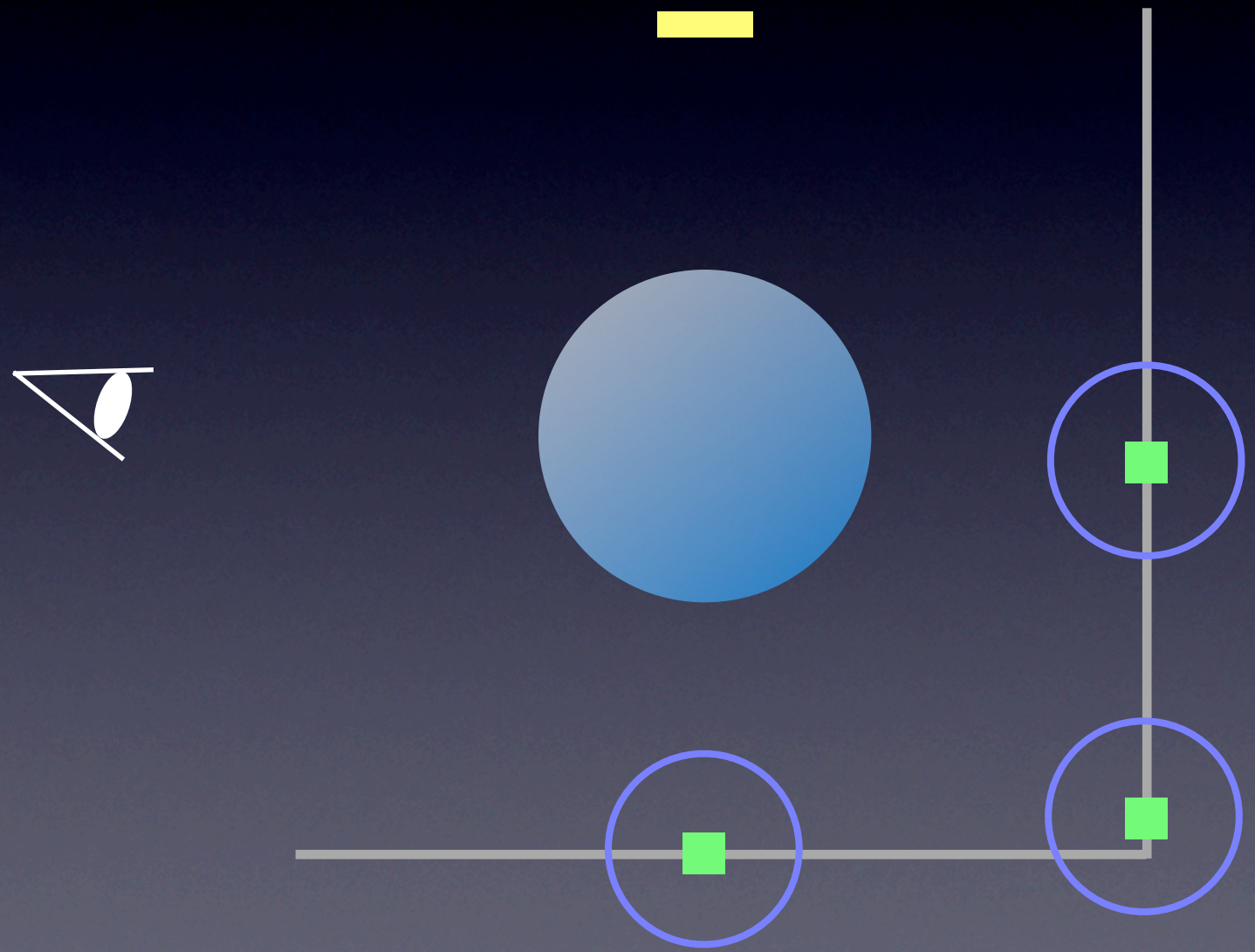




Initial Pass



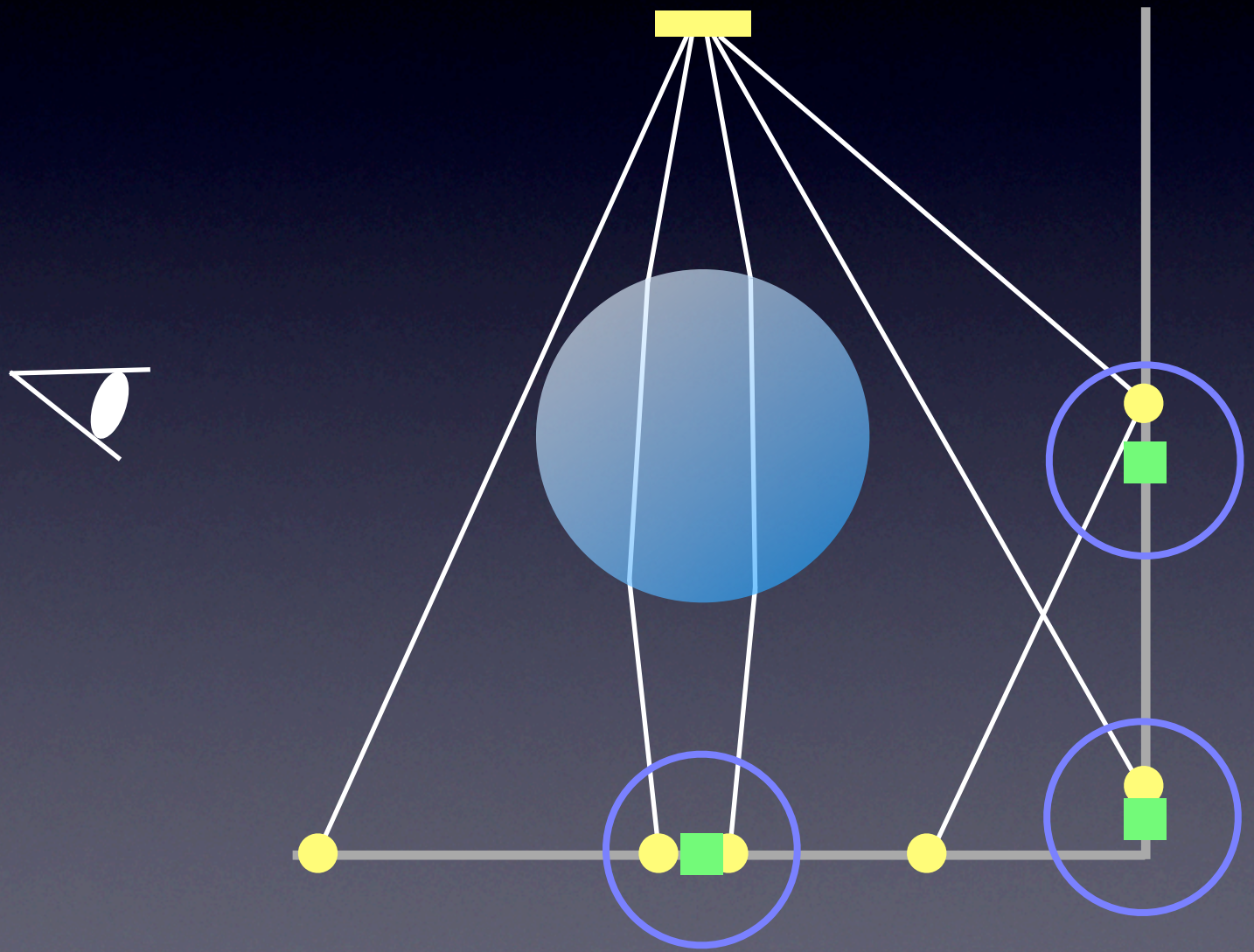
Initial Pass



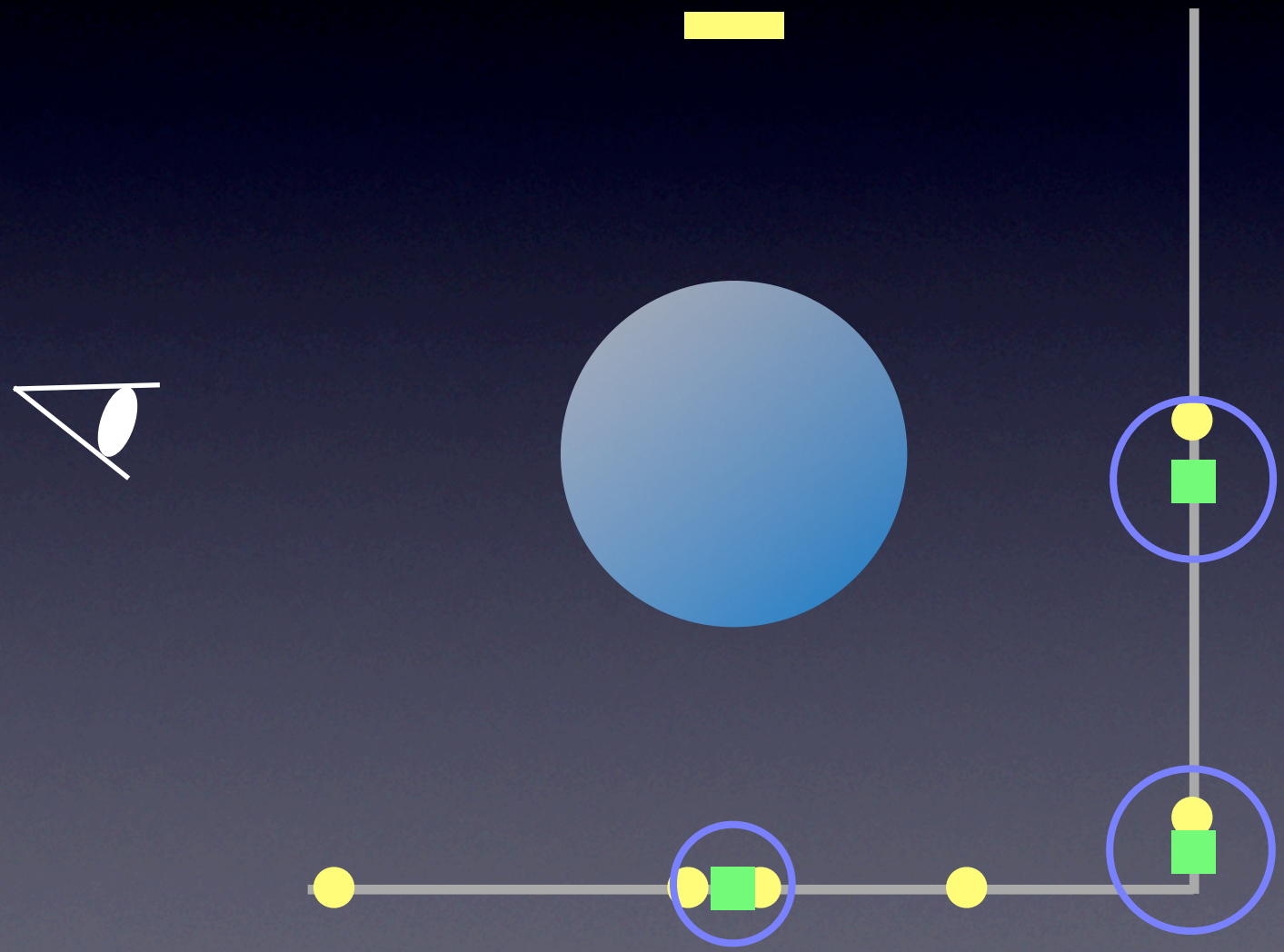
Photon Statistics

- Per intersection point
  - Accumulated flux times BRDF $\tau_i\left(x, \vec{\omega}\right)$
  - Search radius $R_i\left(x\right)$
  - Local photon count $N_i(x)$
- Global
  - Emitted photon count $N_e(i)$

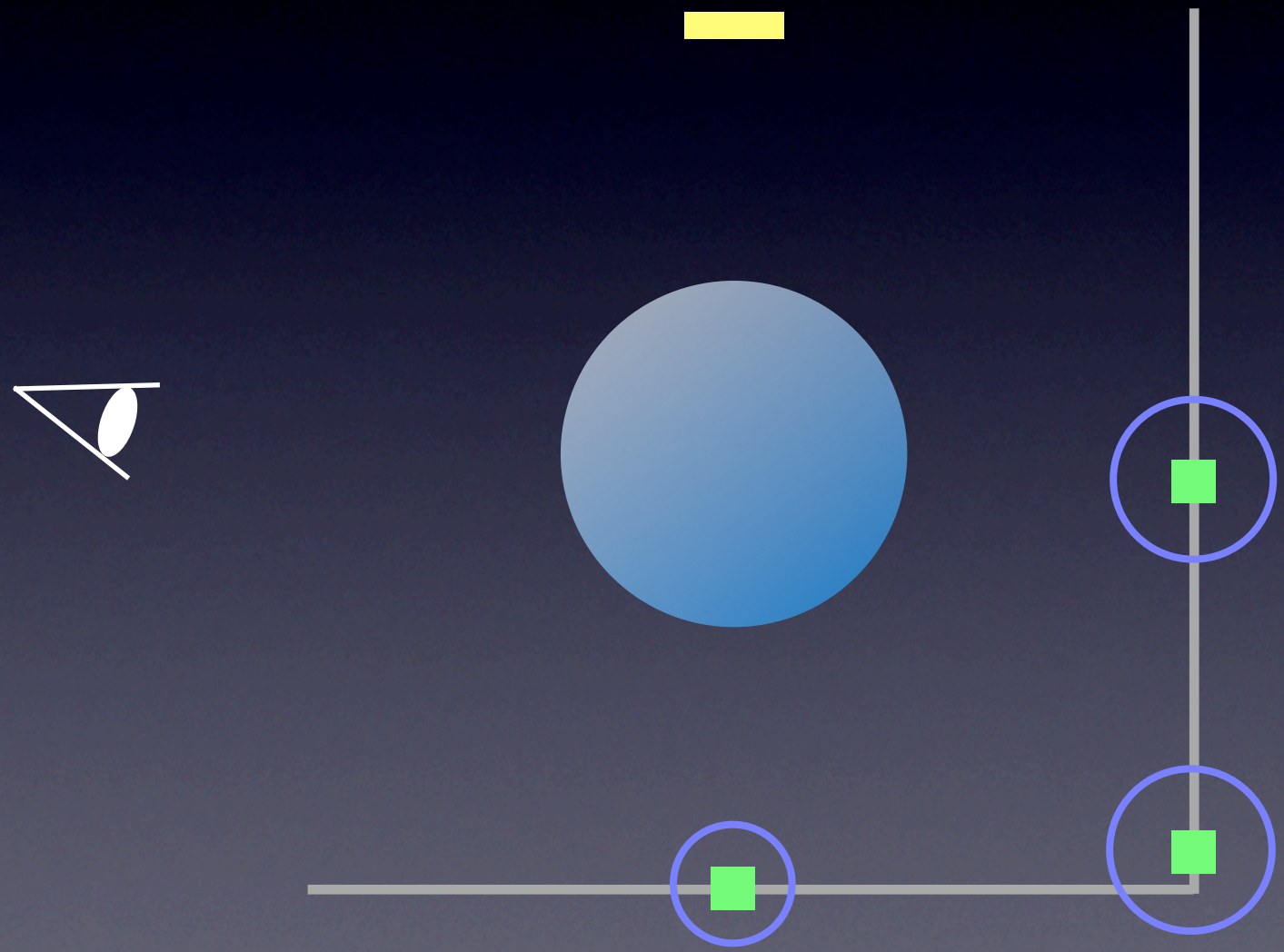
Photon Pass



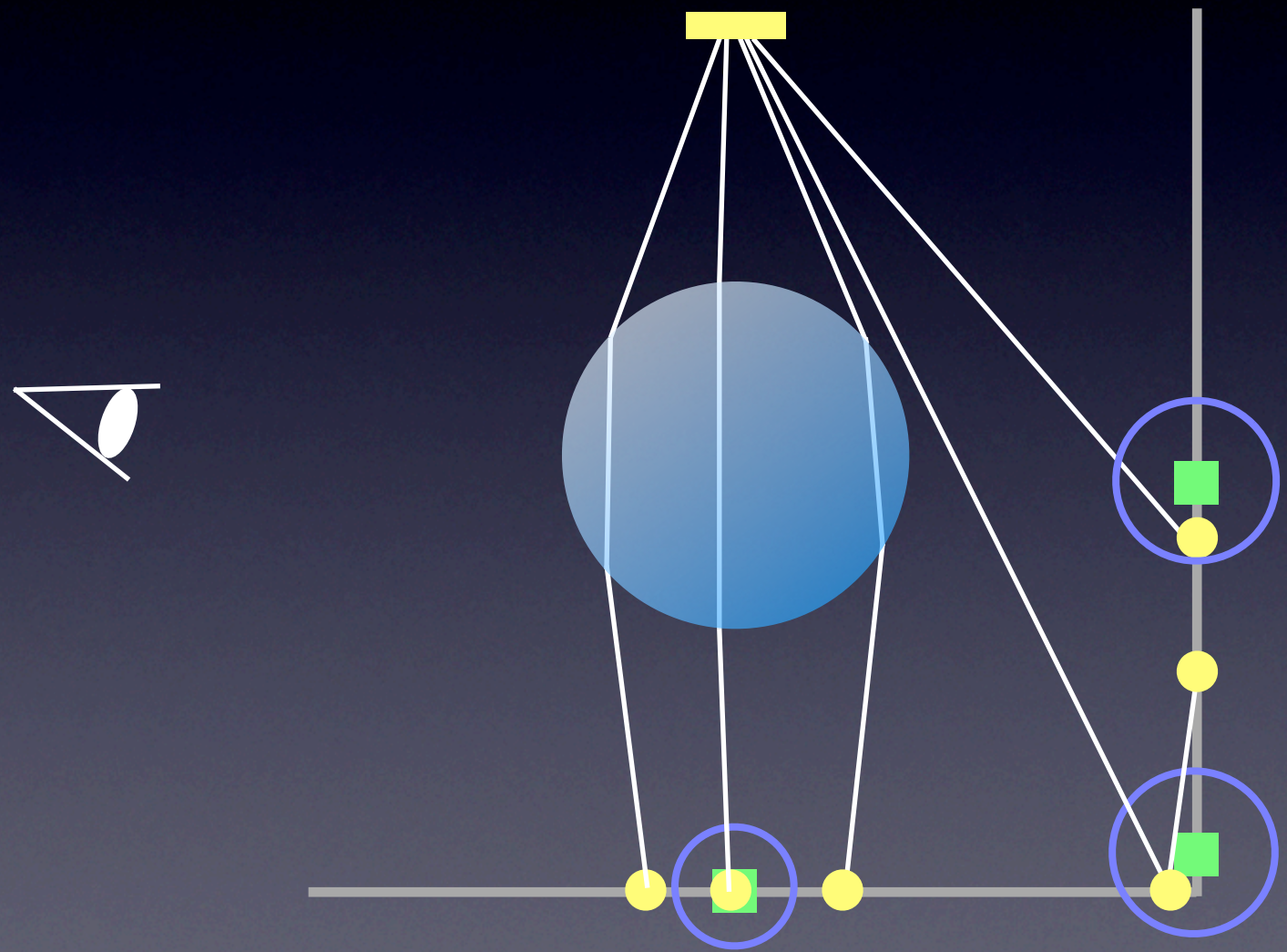
Photon Pass



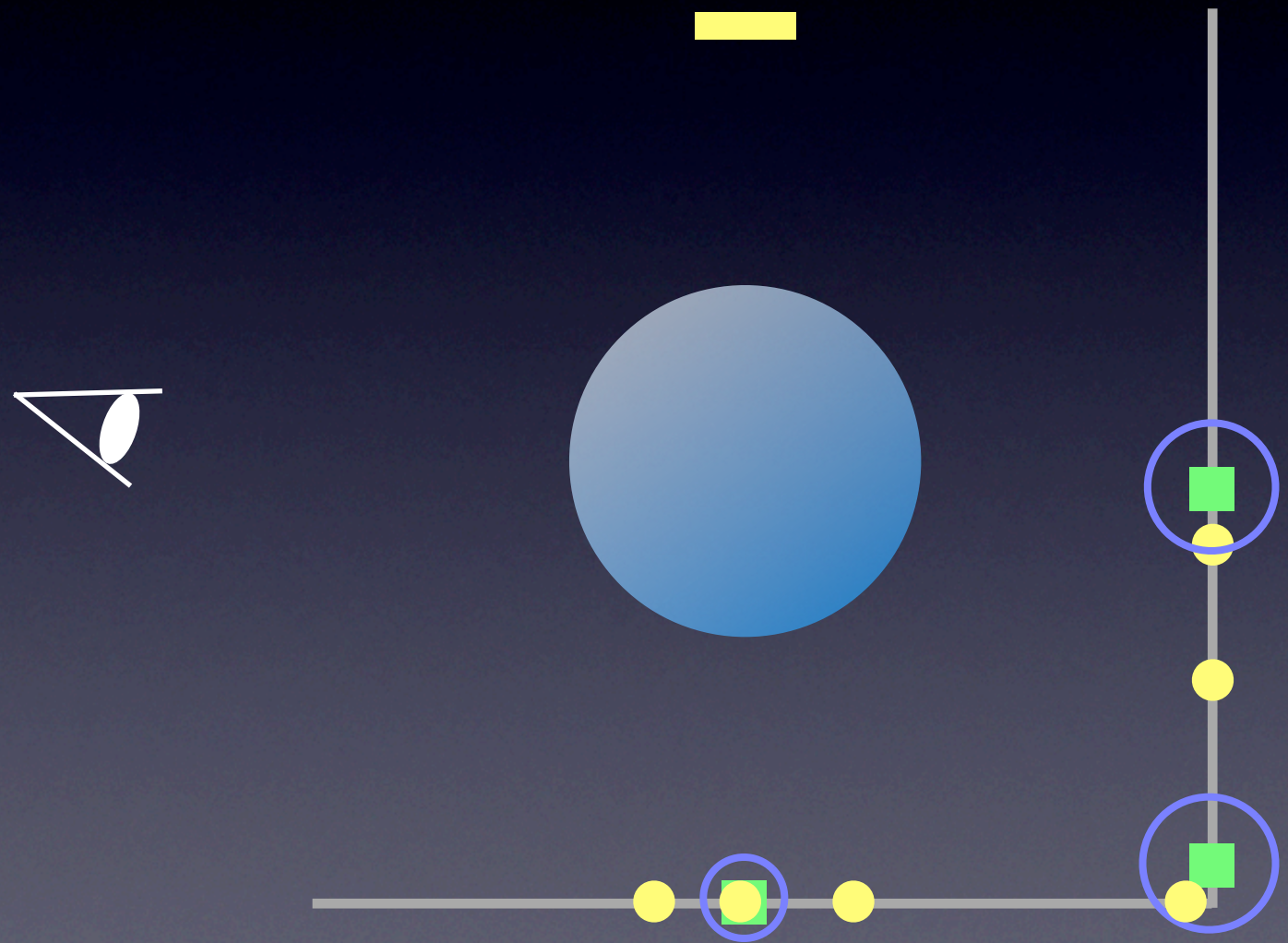
Photon Pass



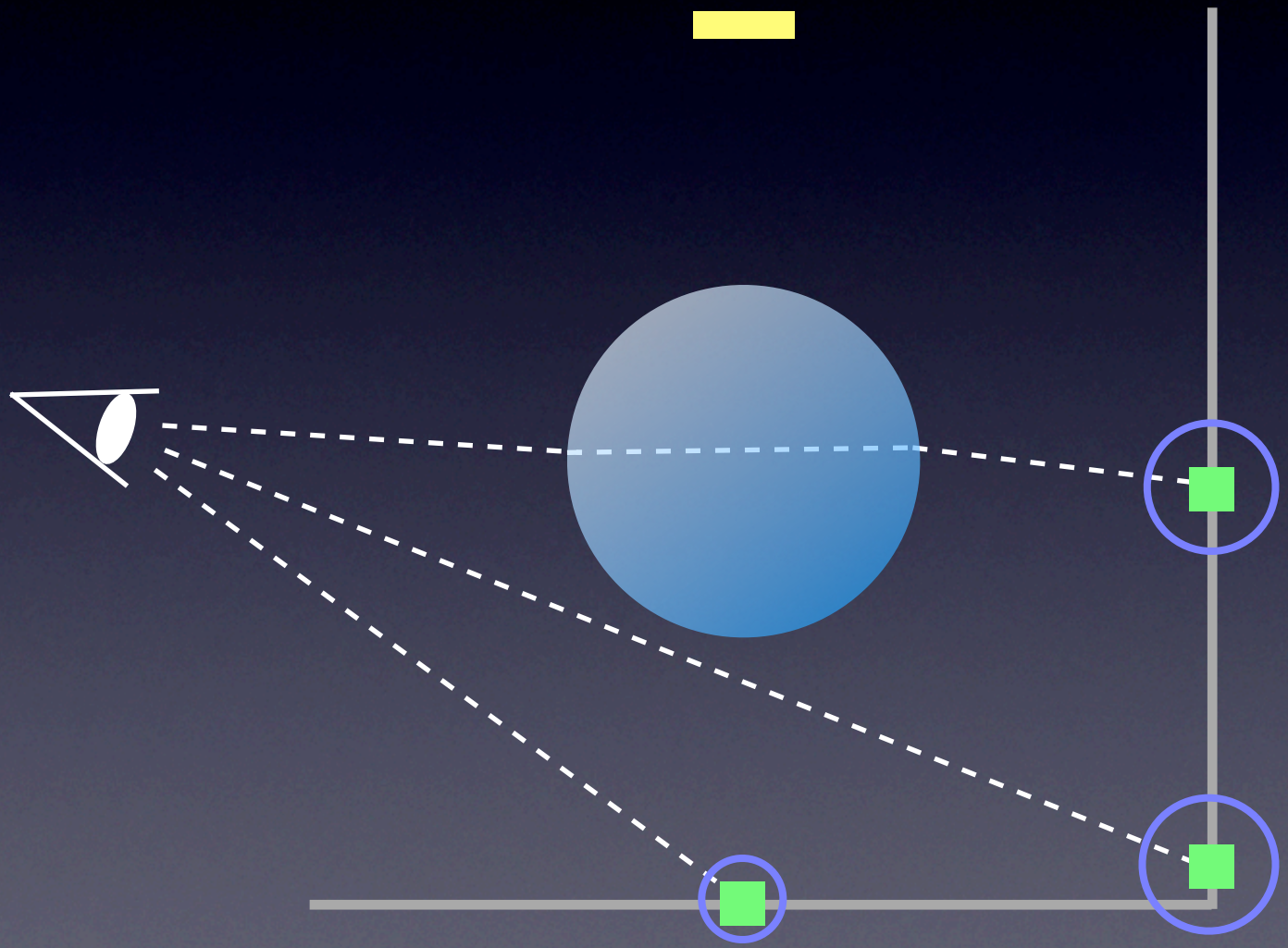
Next Photon Pass



Next Photon Pass



Rendering



Progressive Radiance Estimate

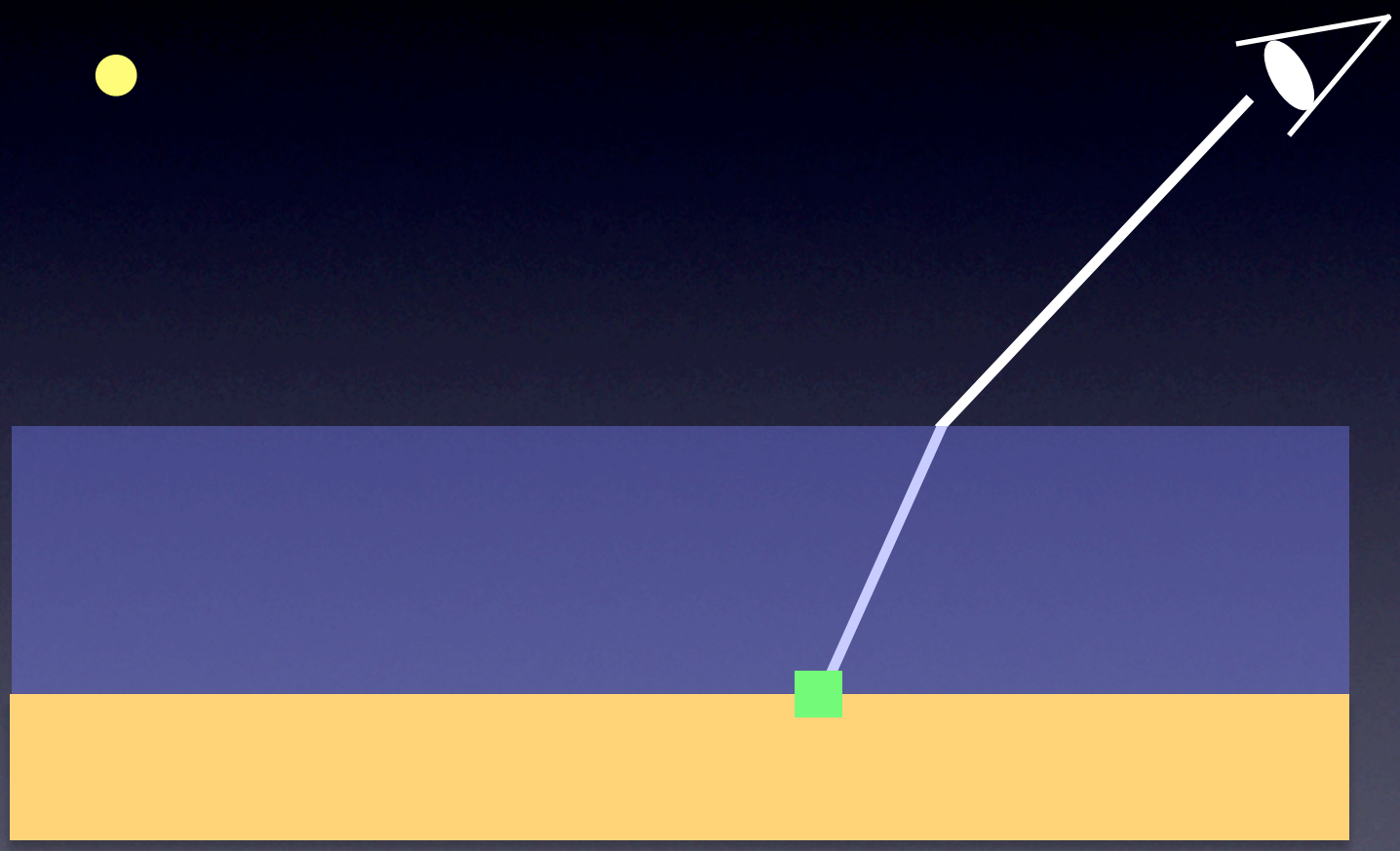
$$L_i\left(x,\vec{\omega}\right) = \frac{\tau_i\left(x,\vec{\omega}\right)}{\pi R_i\left(x\right)^2 N_e(i)}$$

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

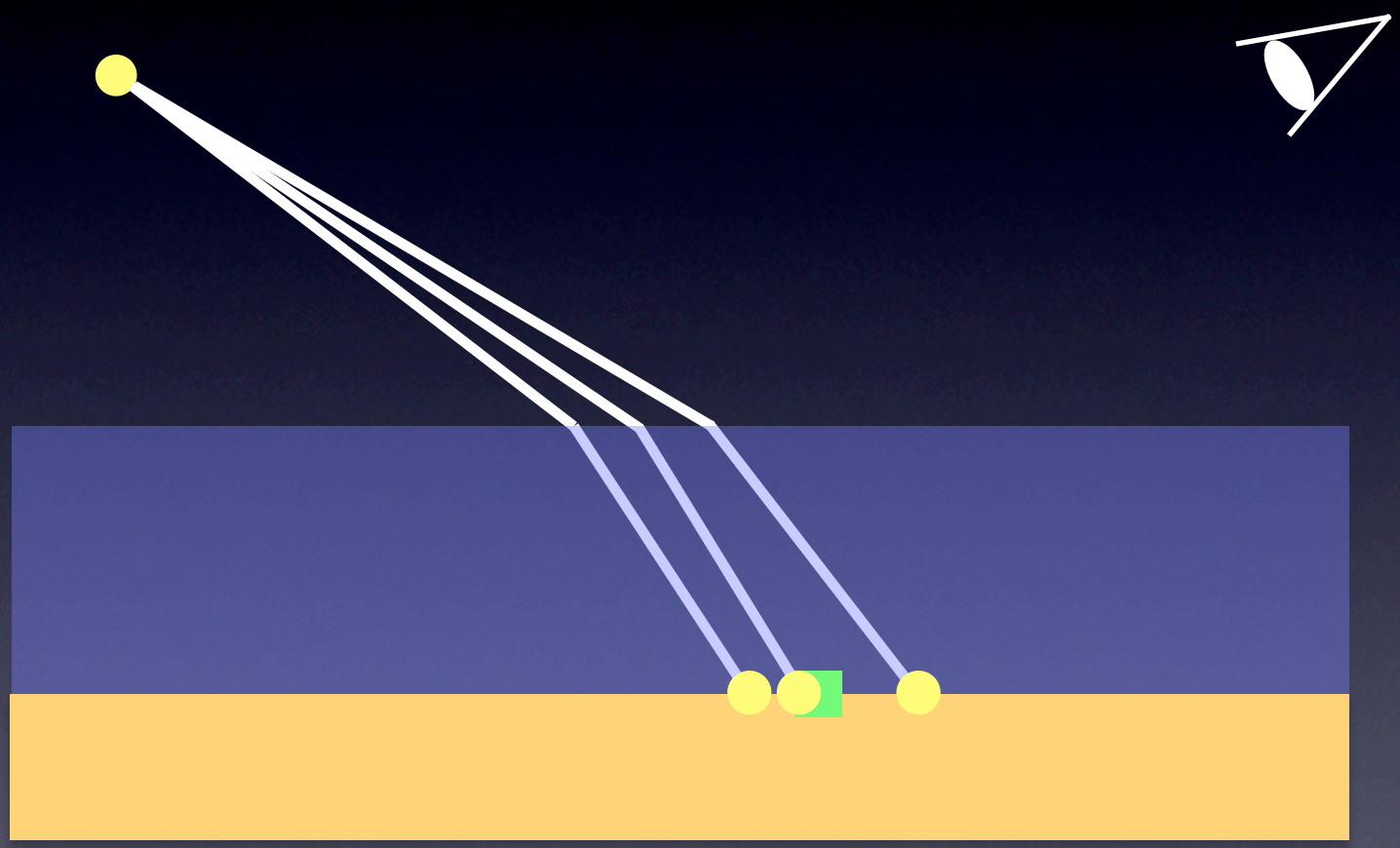
Stochastic Progressive Photon Mapping

Motivation

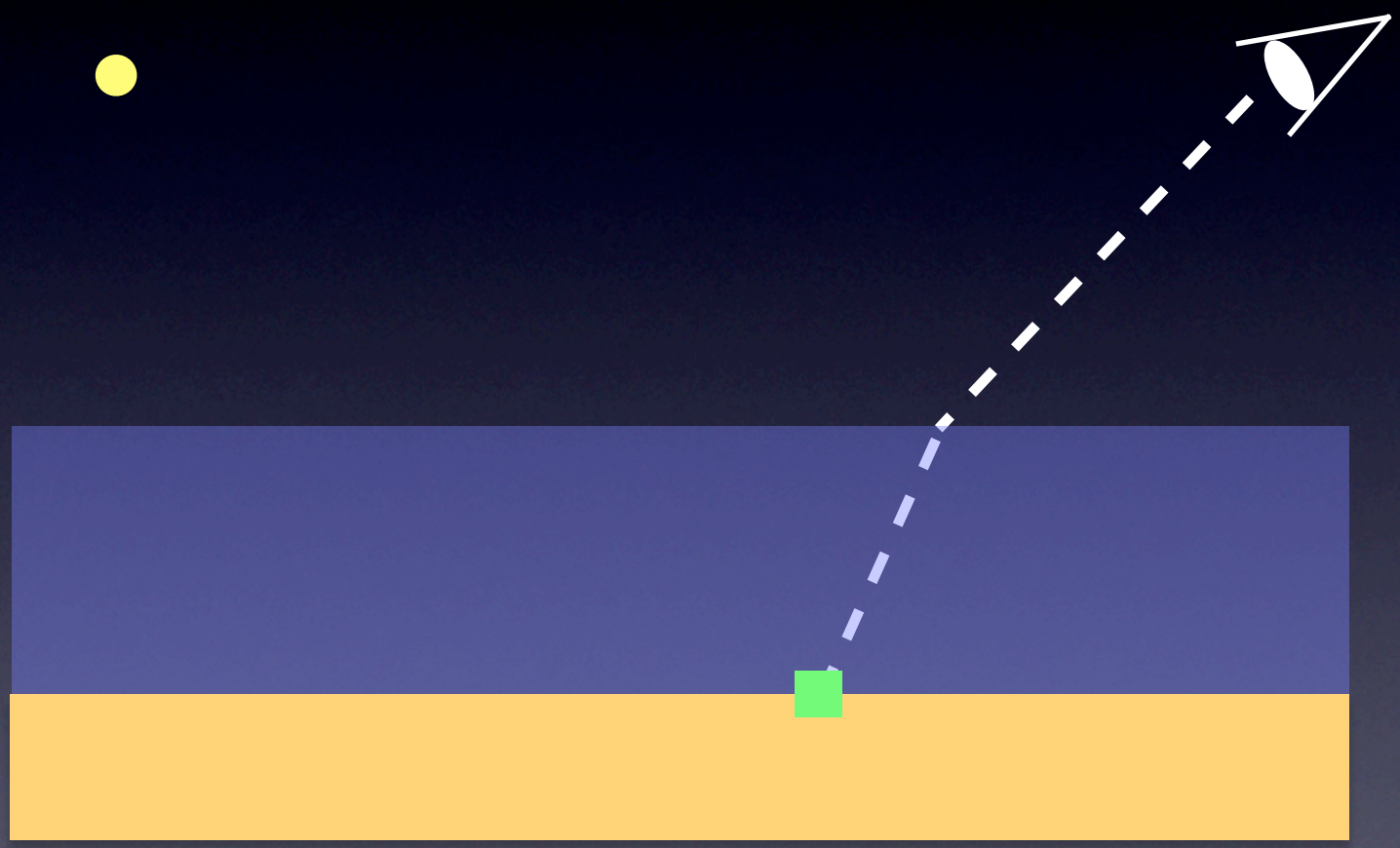
Initial Pass



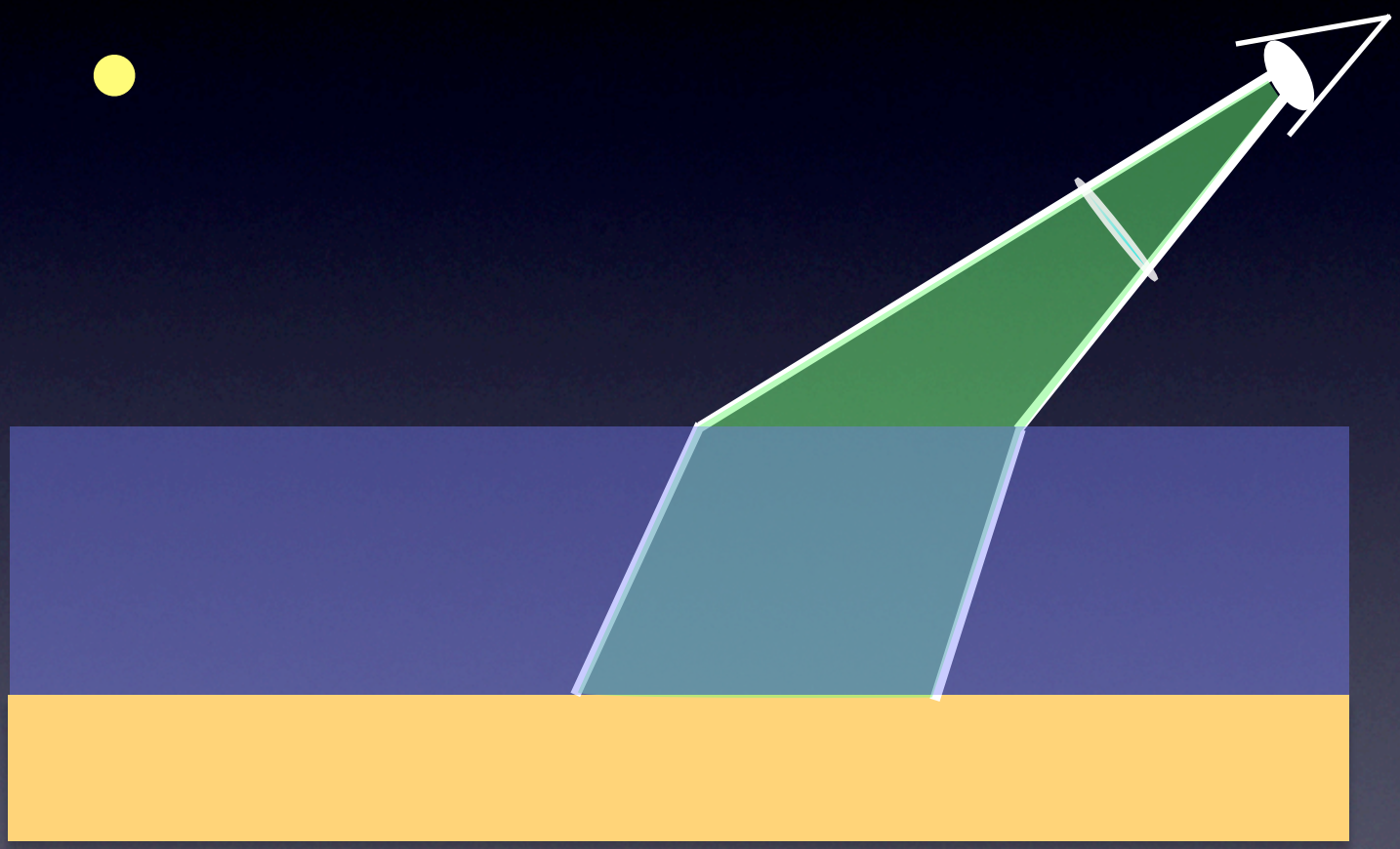
Photon Pass



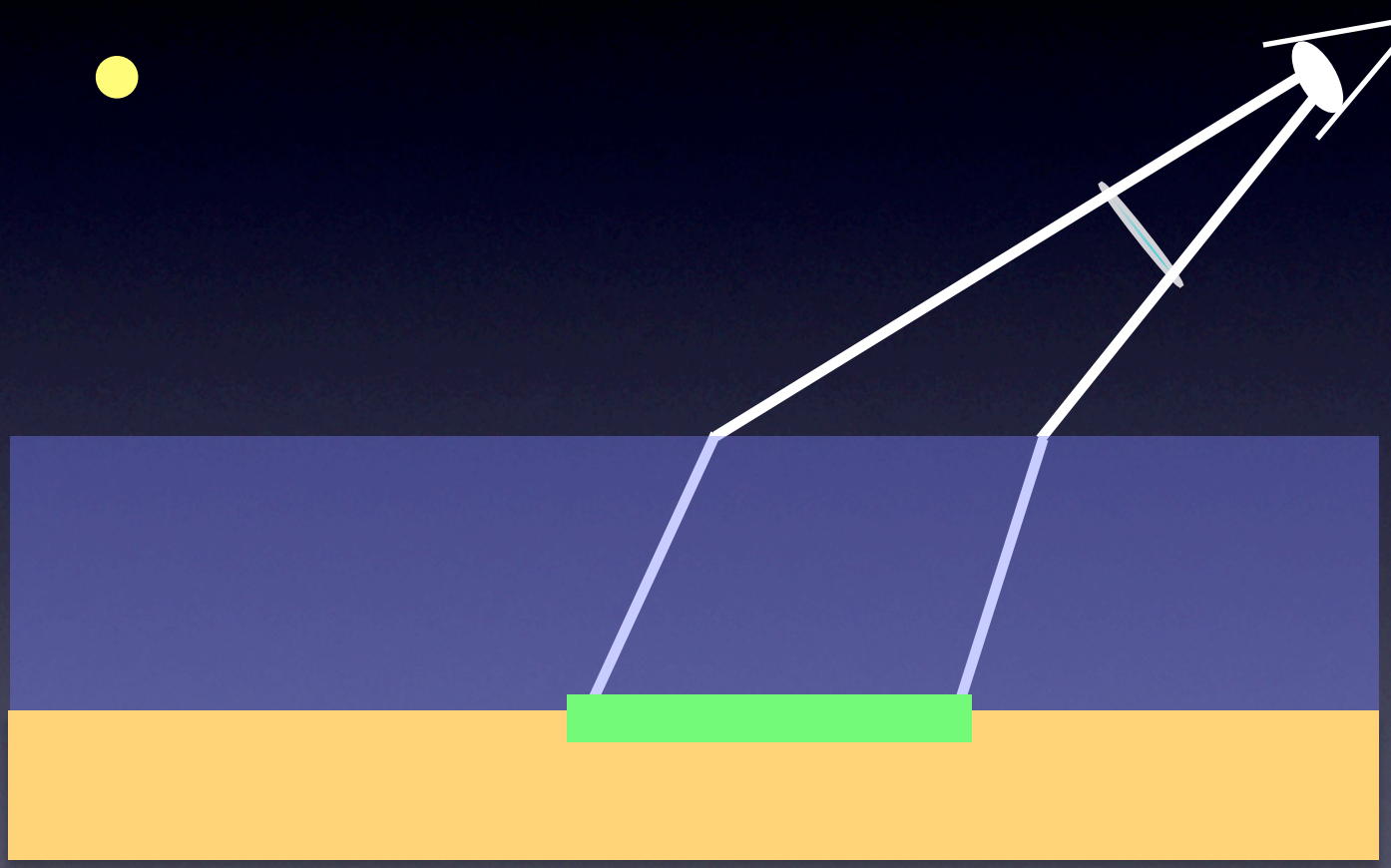
Rendering



Non-zero Aperture



Non-zero Aperture

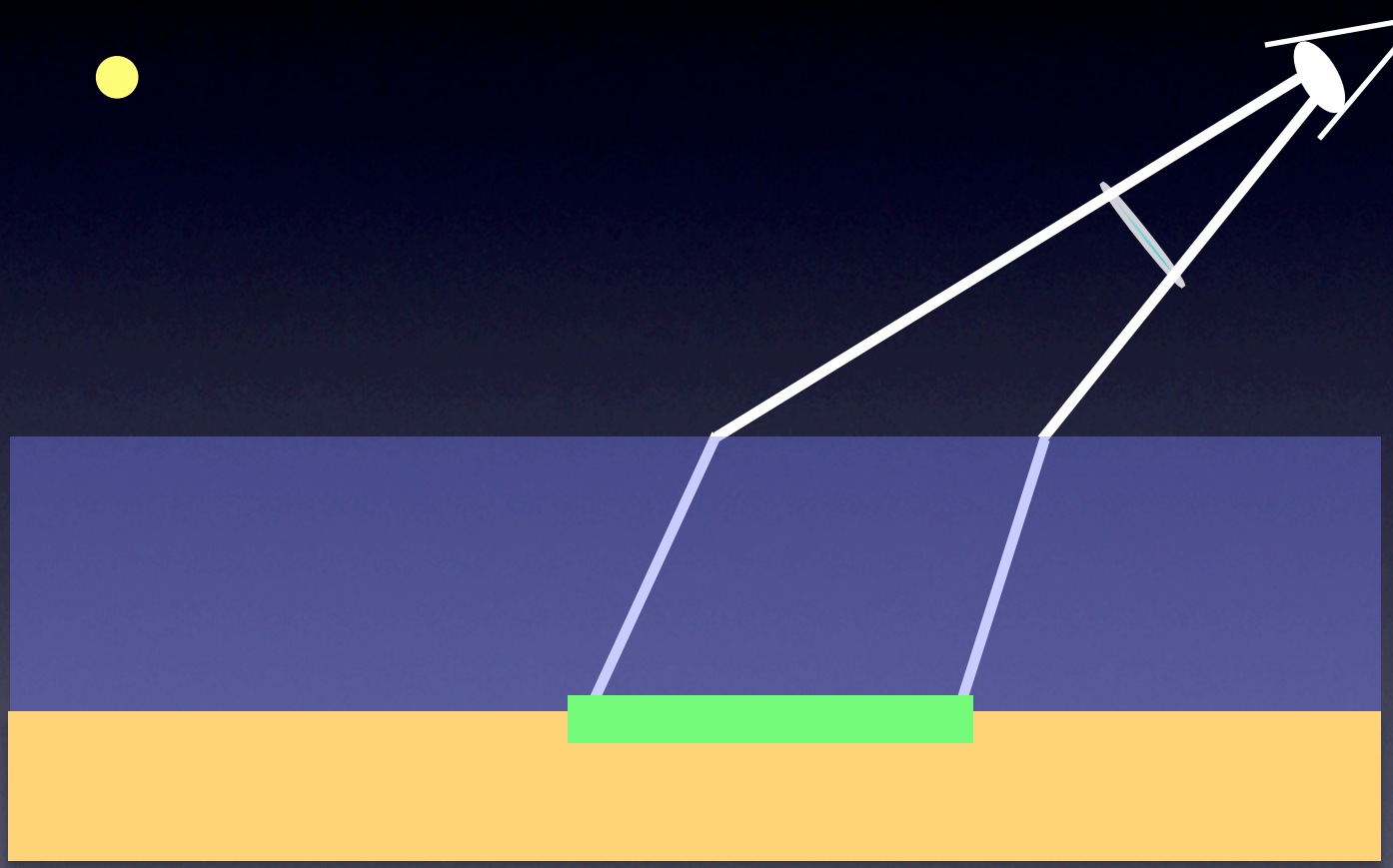


Point-wise Radiance Estimate

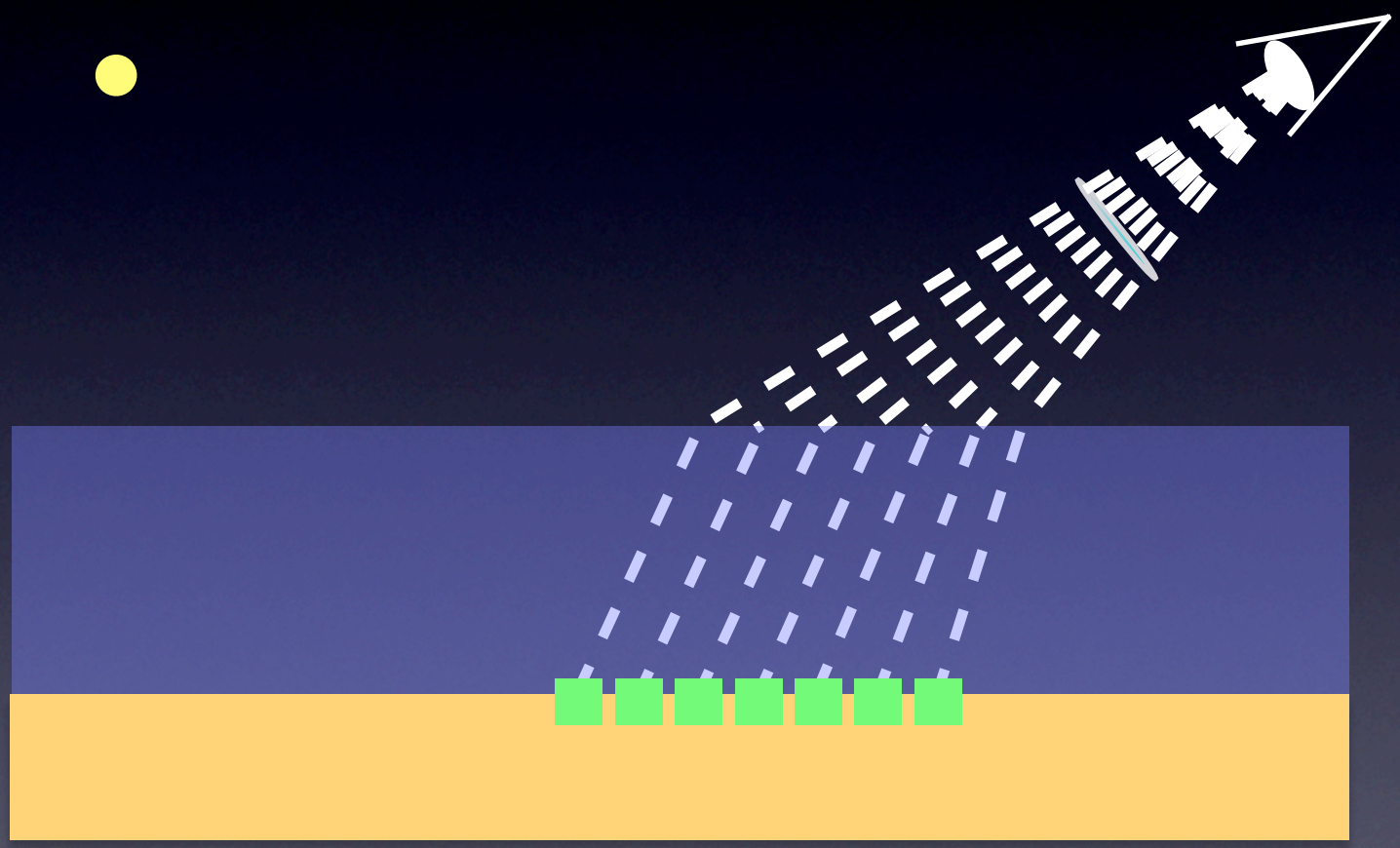
- Converges to the correct radiance at a point

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

Non-zero Aperture with PPM

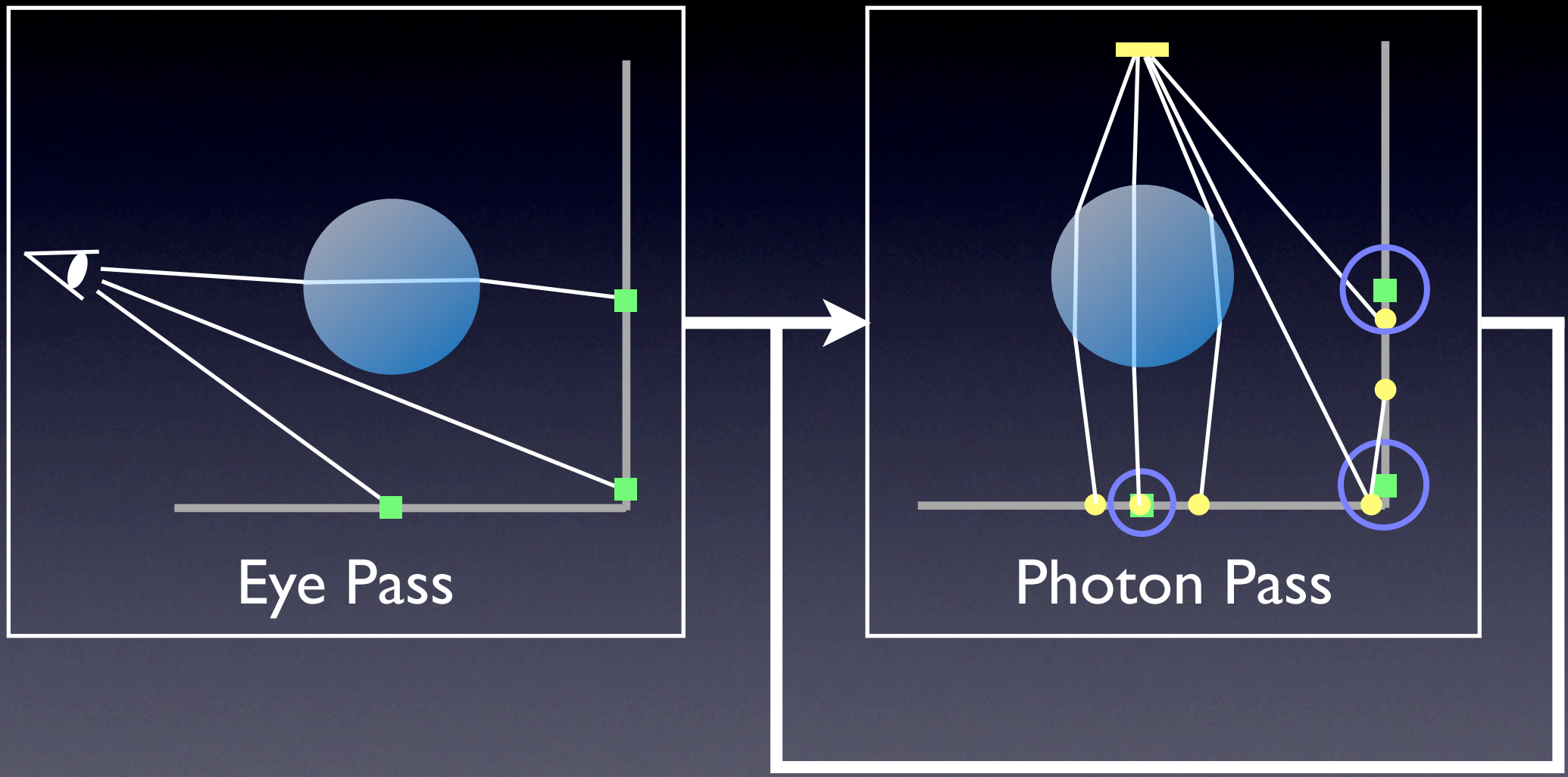


Non-zero Aperture with PPM

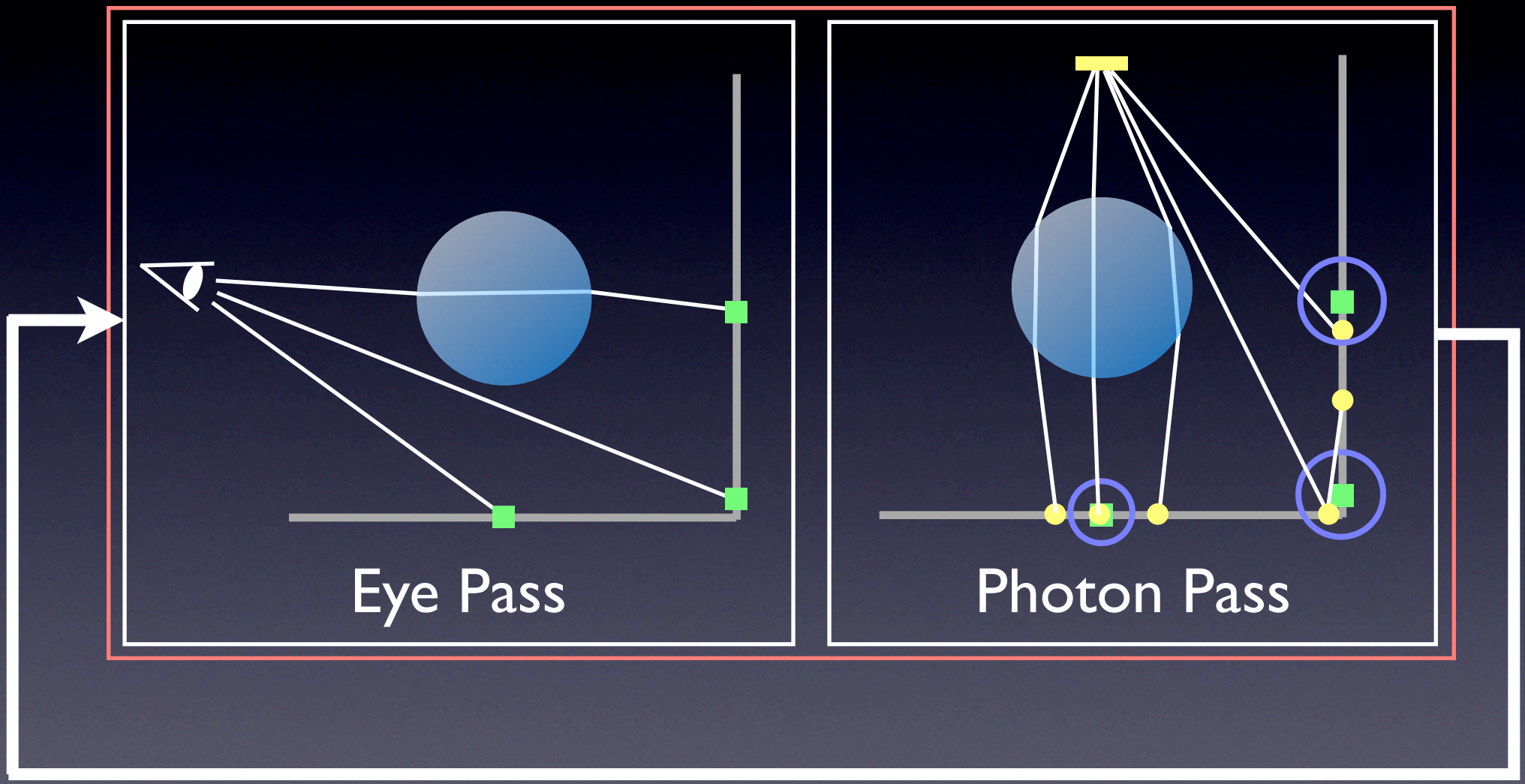


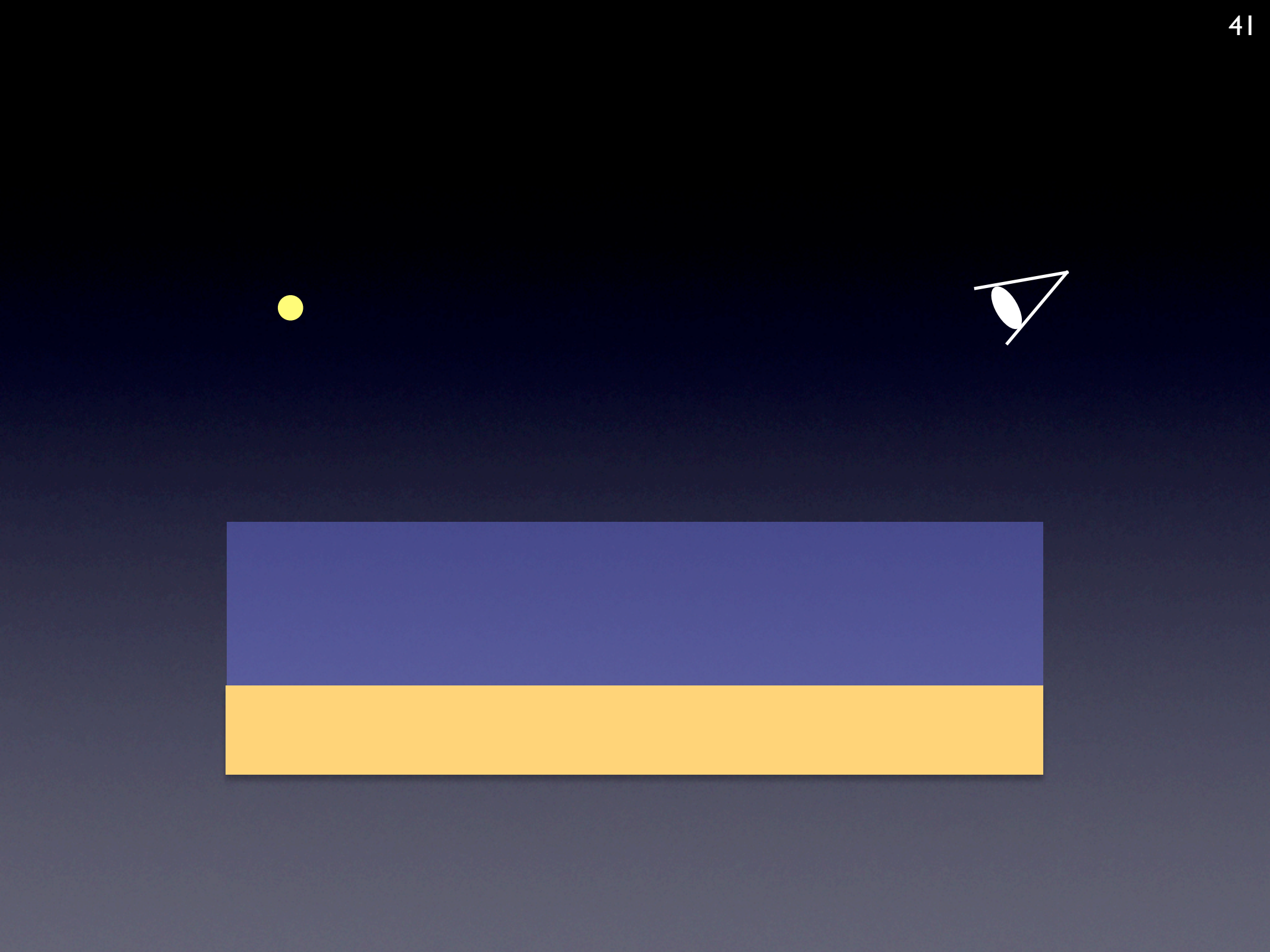
Method

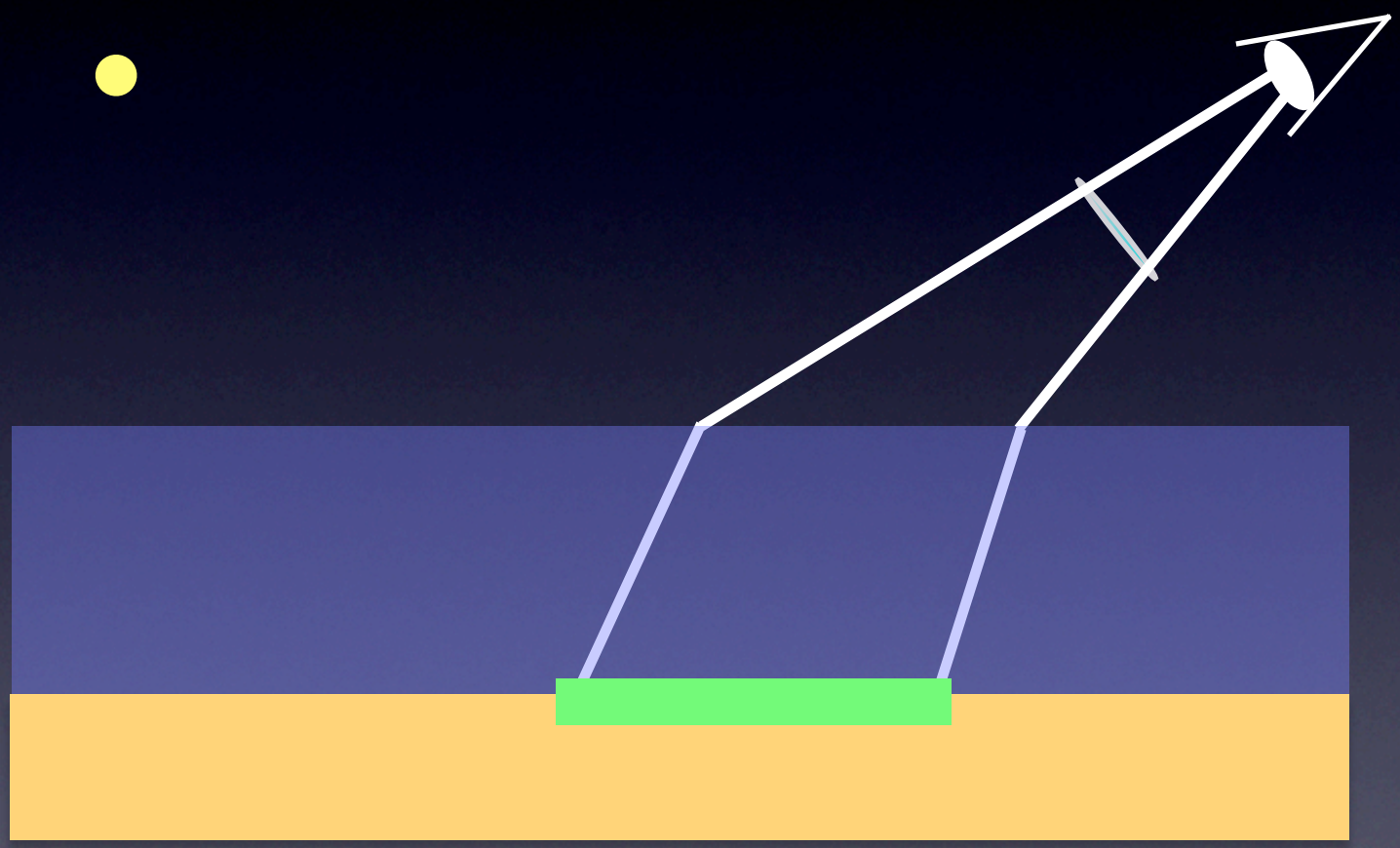
Key Idea



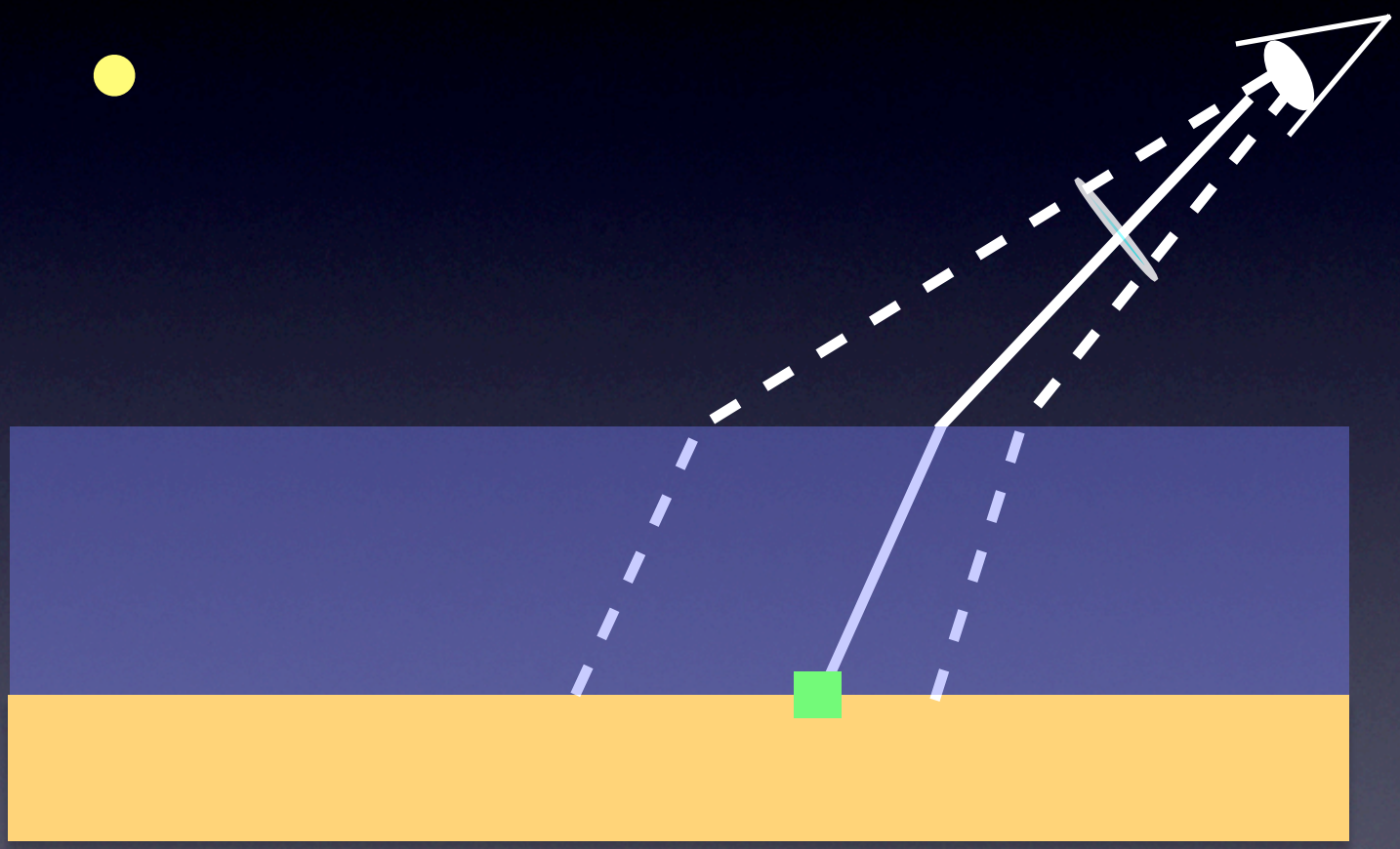
Key Idea



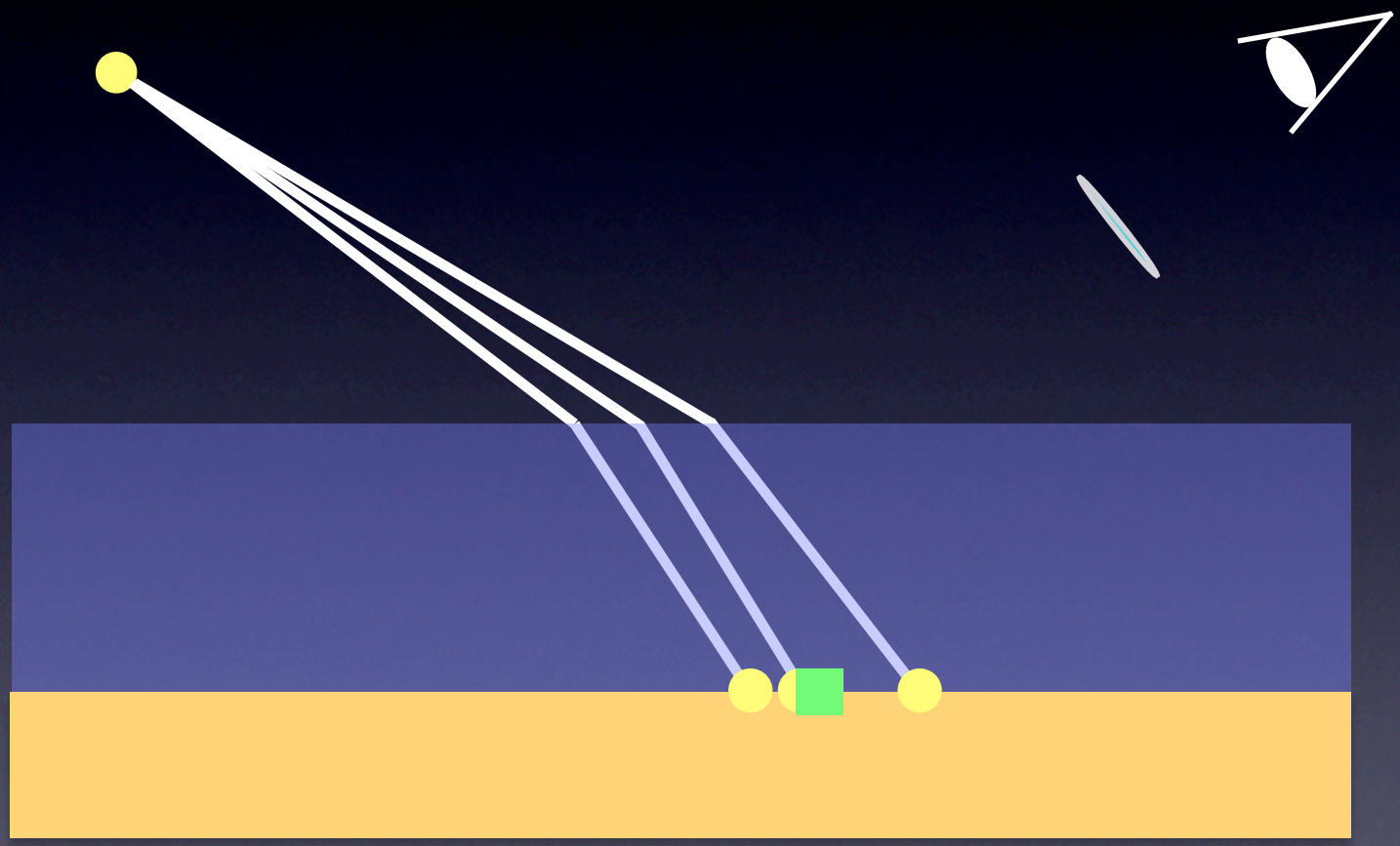




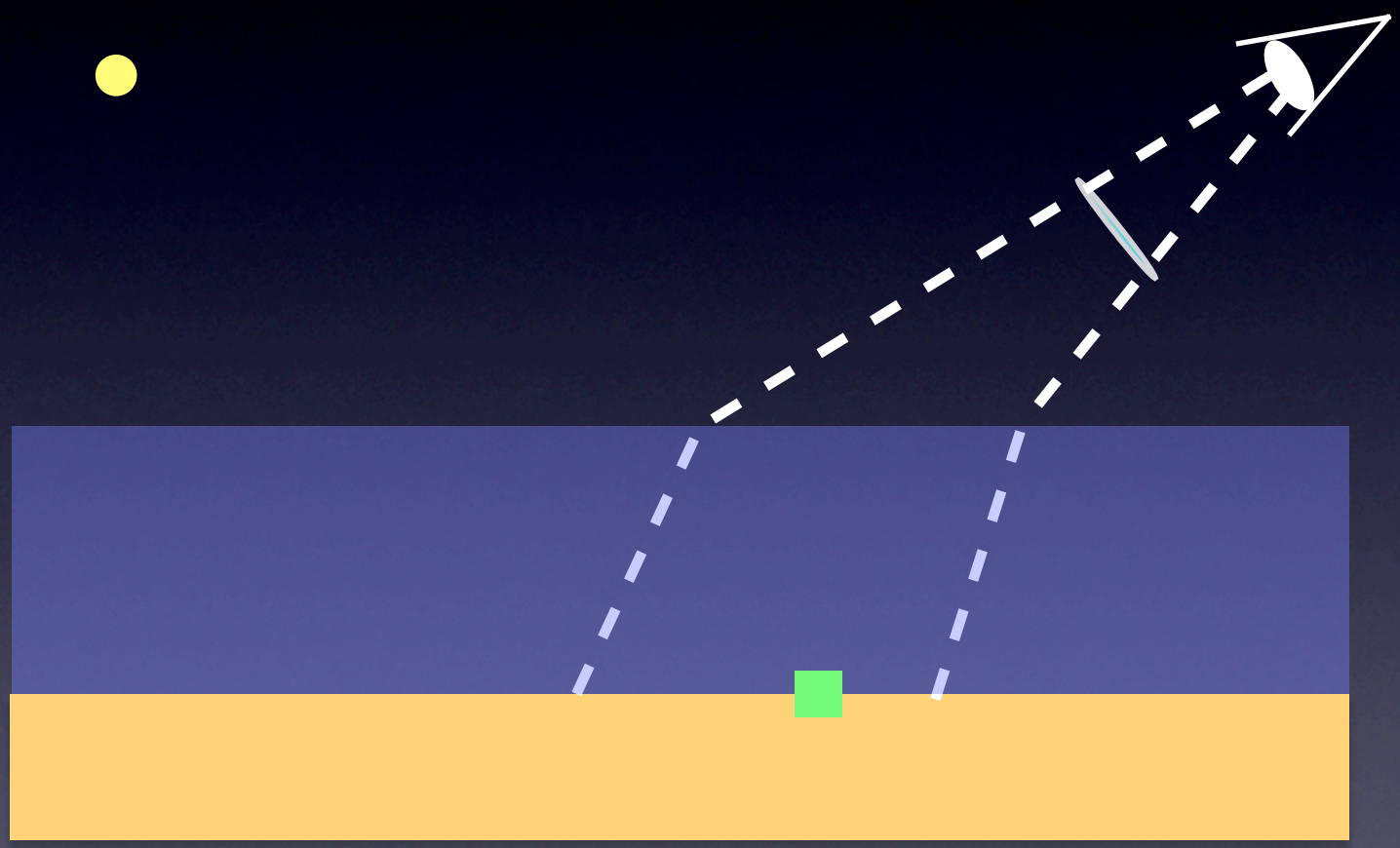
Eye Pass



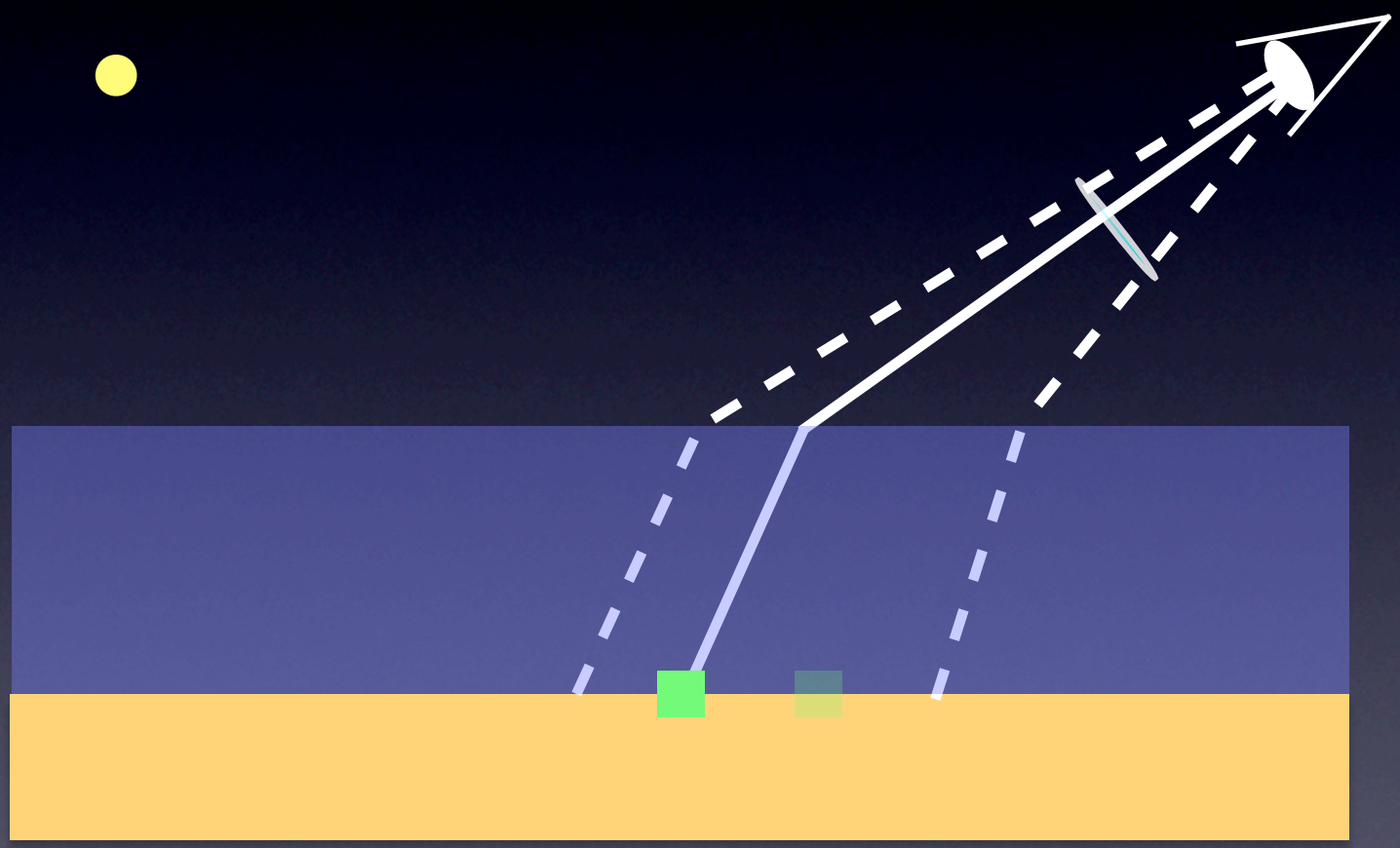
Photon Pass



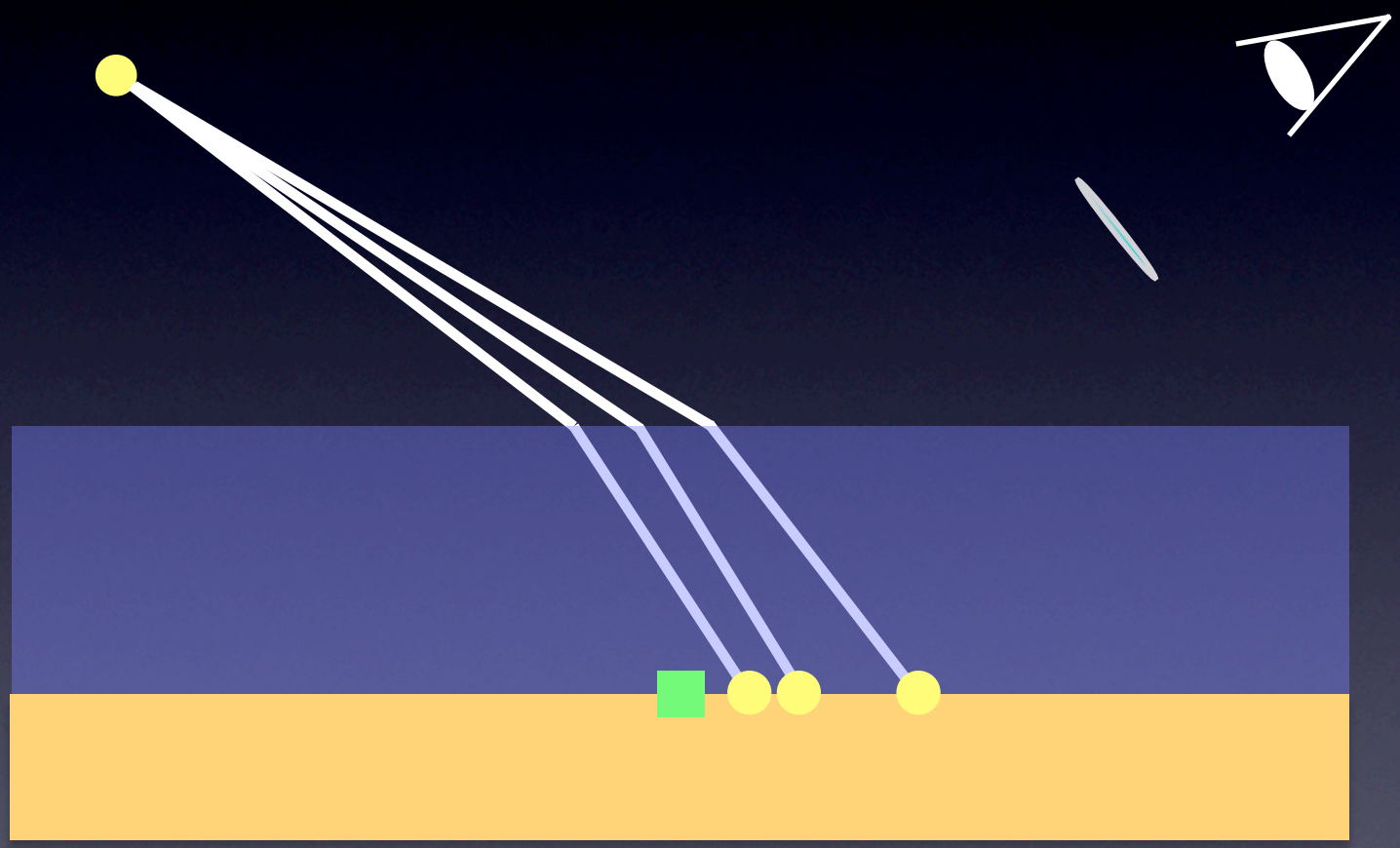
Next Eye Pass



Next Eye Pass



Next Photon Pass



Photon Statistics

- Per intersection point x
  - Accumulated flux times BRDF $\tau_i\left(\boxed{x}, \vec{\omega}\right)$
  - Search radius $R_i\left(\boxed{x}\right)$
  - Local photon count $N_i\left(\boxed{x}\right)$

- Global
  - Emitted photon count $N_e(i)$

Average Photon Statistics

- Per region S
  - Accumulated flux times BRDF $\tau_i\left(S, \vec{\omega}\right)$
  - Search radius $R_i\left(S\right)$
  - Local photon count $N_i\left(S\right)$
- Global
  - Emitted photon count $N_e(i)$

Progressive Radiance Estimate

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

Stochastic Progressive Radiance Estimate

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

Stochastic Progressive Radiance Estimate

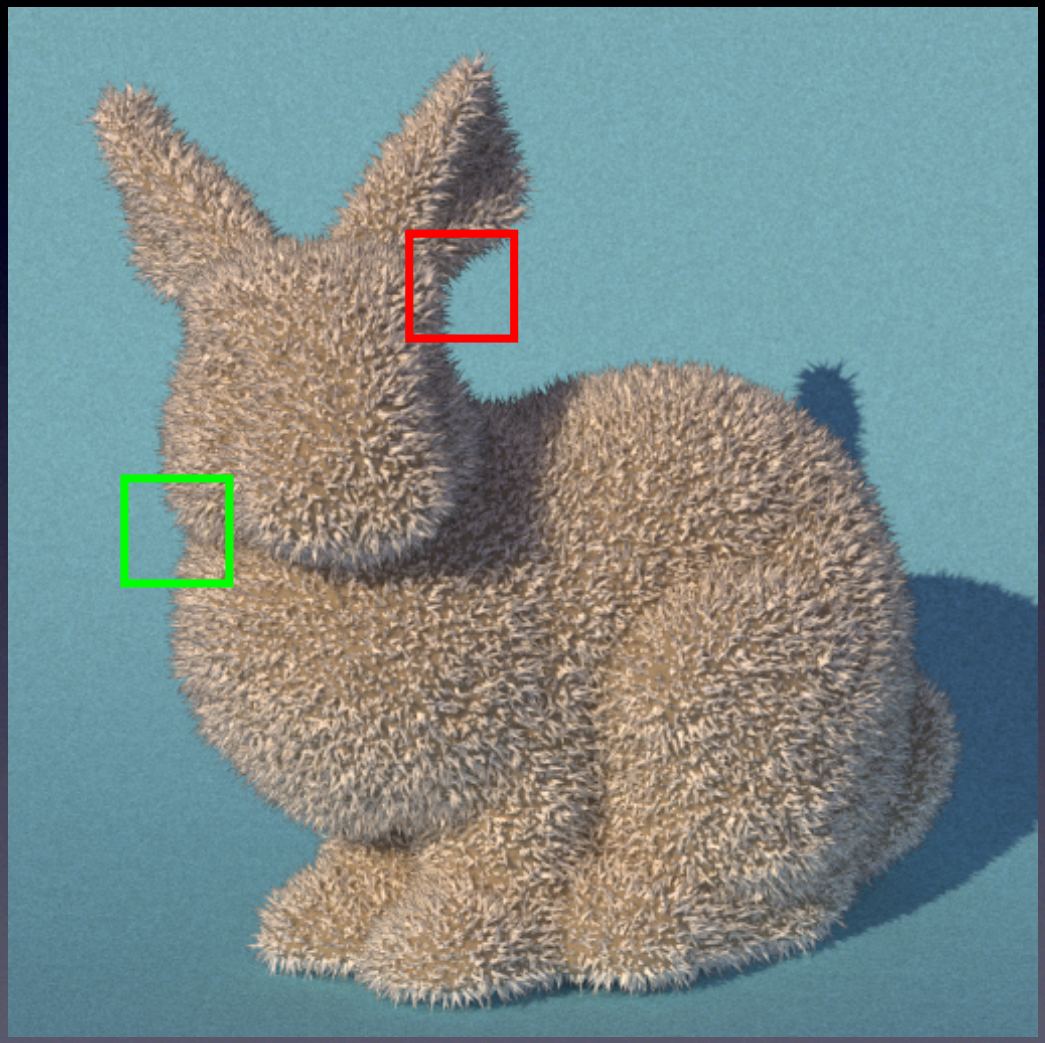
$$L_i\left(S,\vec{\omega}\right) = \frac{\tau_i\left(S,\vec{\omega}\right)}{\pi R_i\left(S\right)^2 N_e(i)}$$

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

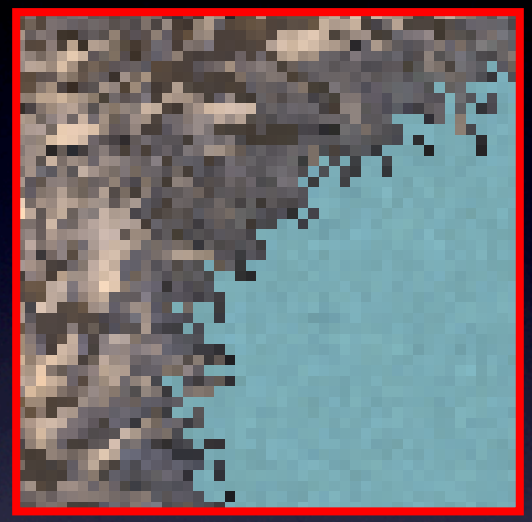
correct average radiance over S

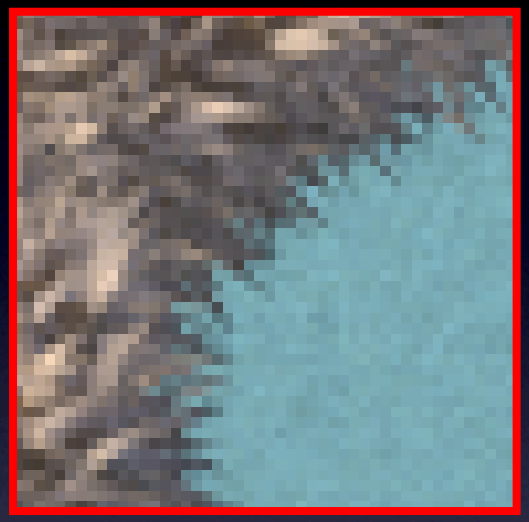
Results

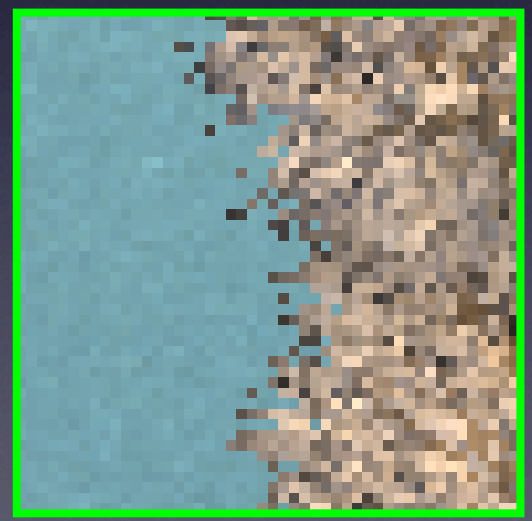
Anti-aliasing



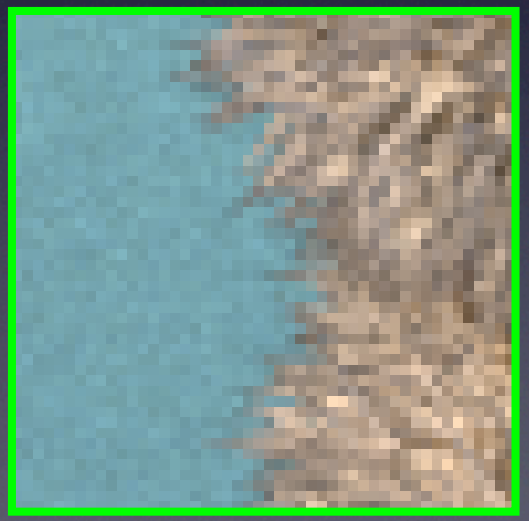
Equal time, Equal memory





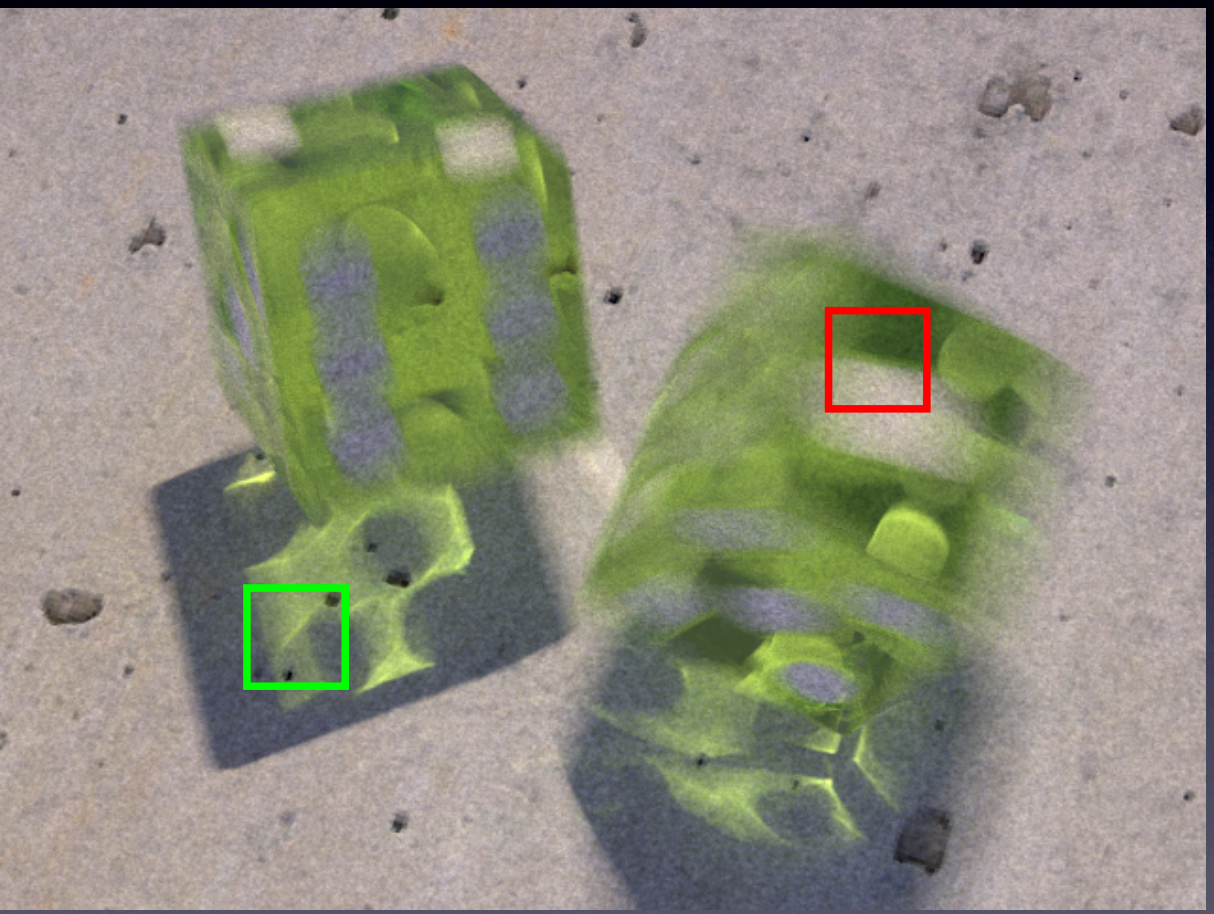


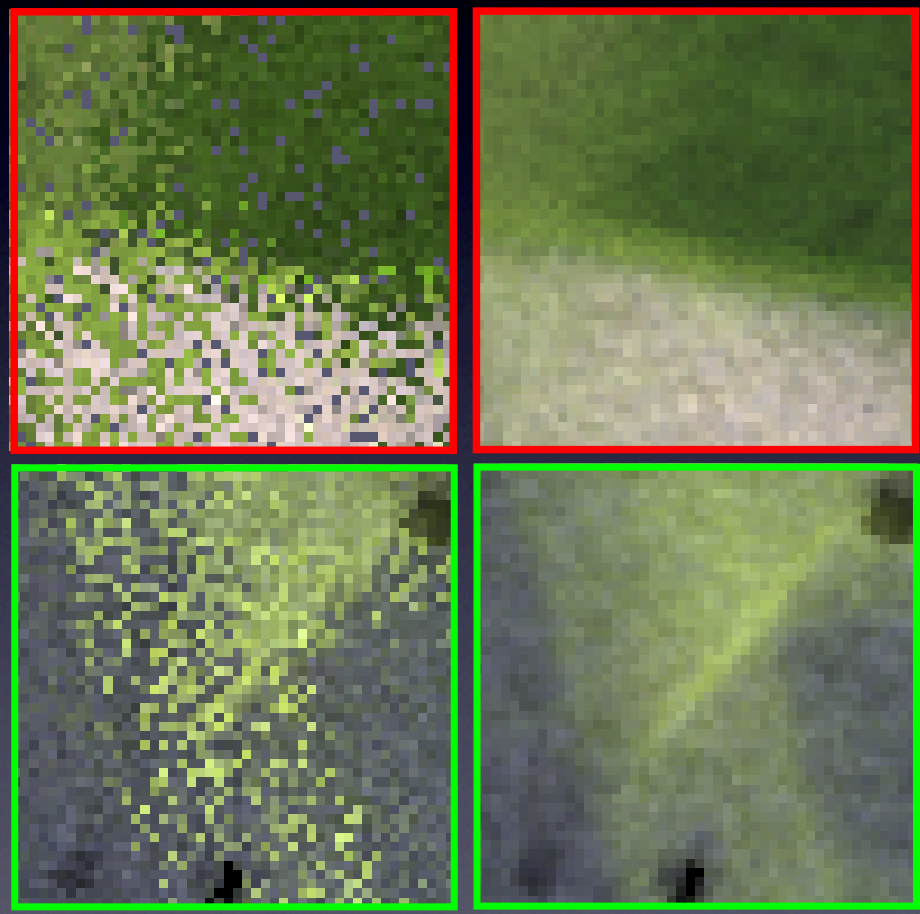
PPM



SPPM

Motion Blur



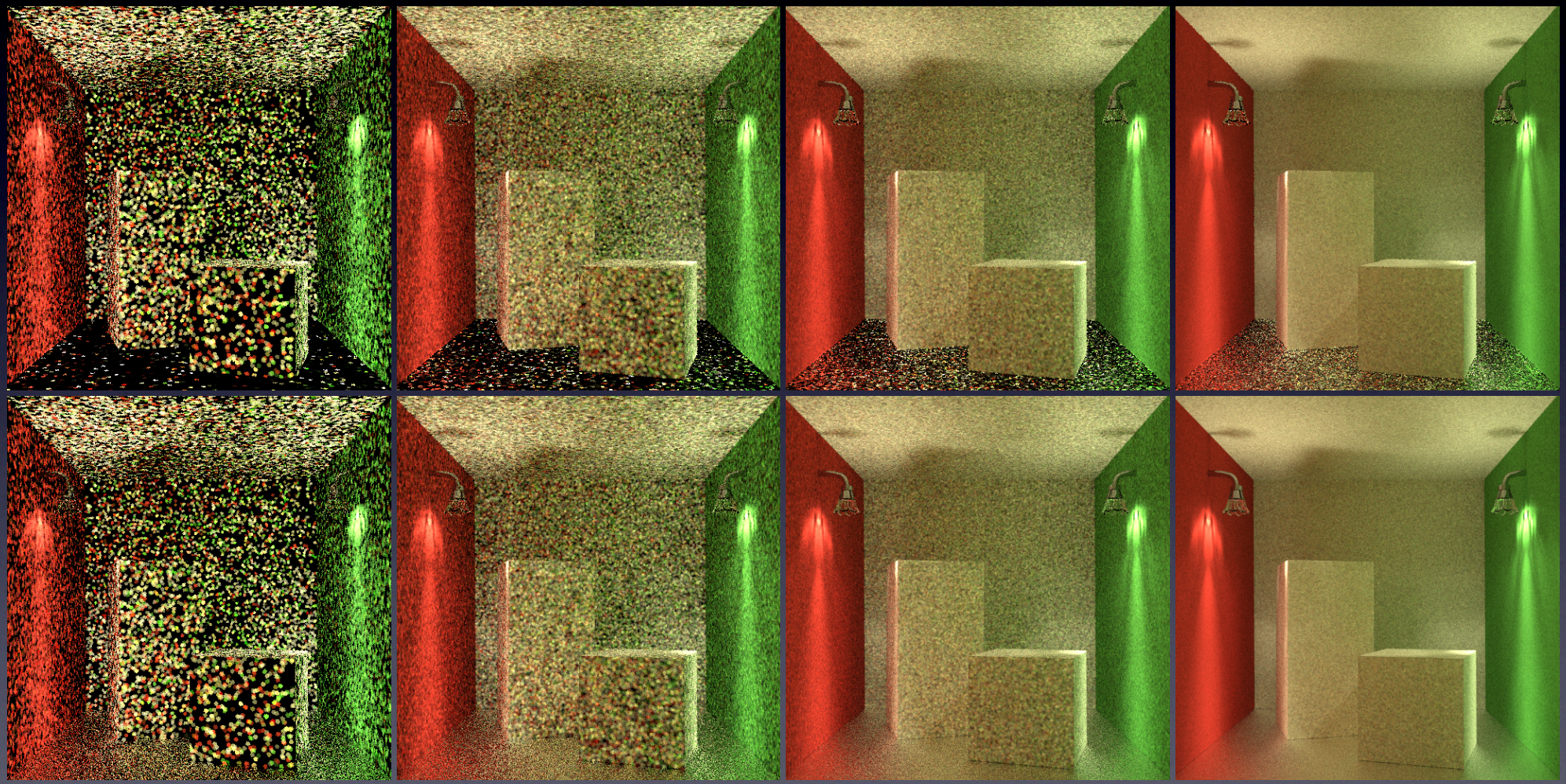


Equal time, Equal memory

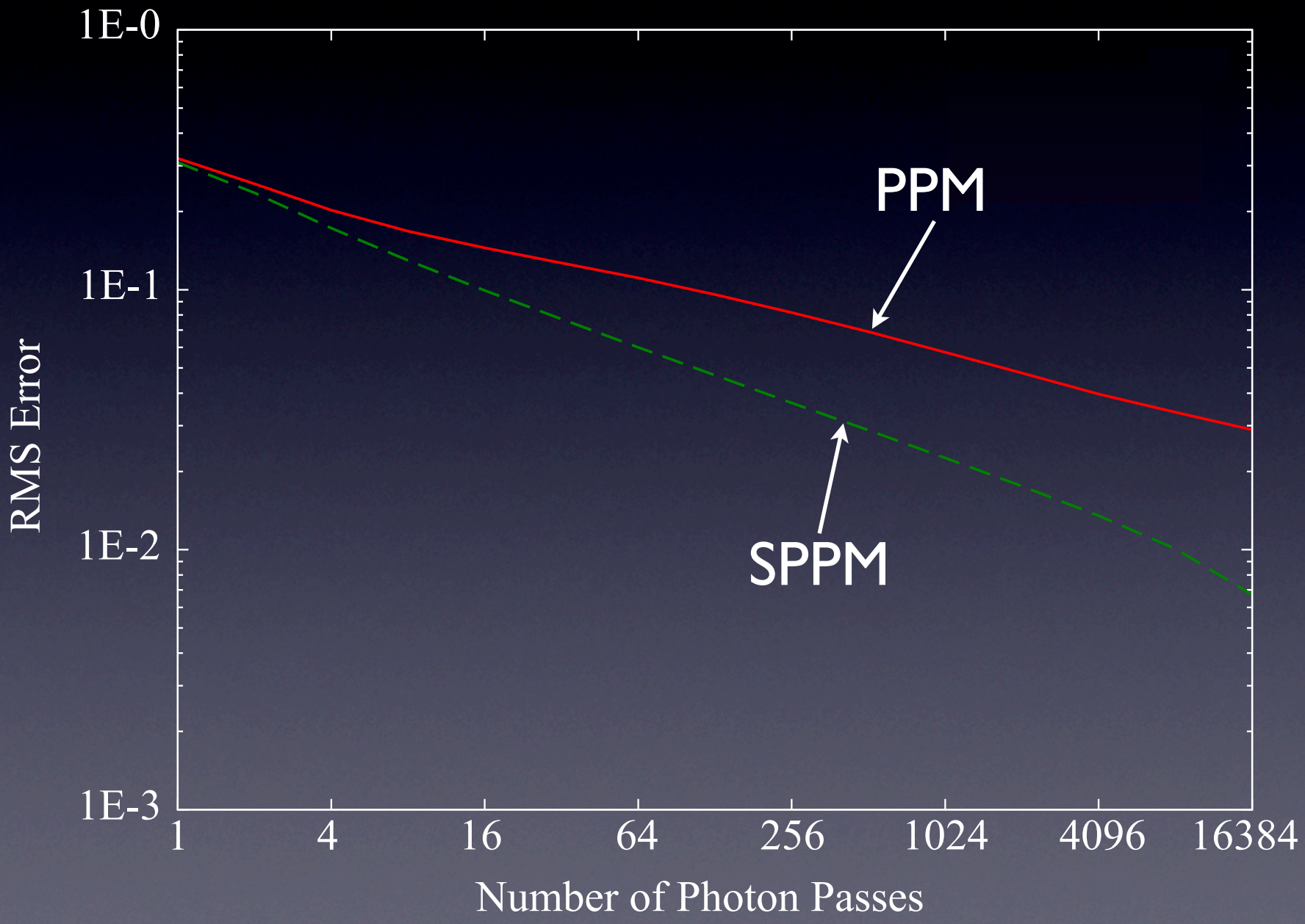
PPM

SPPM

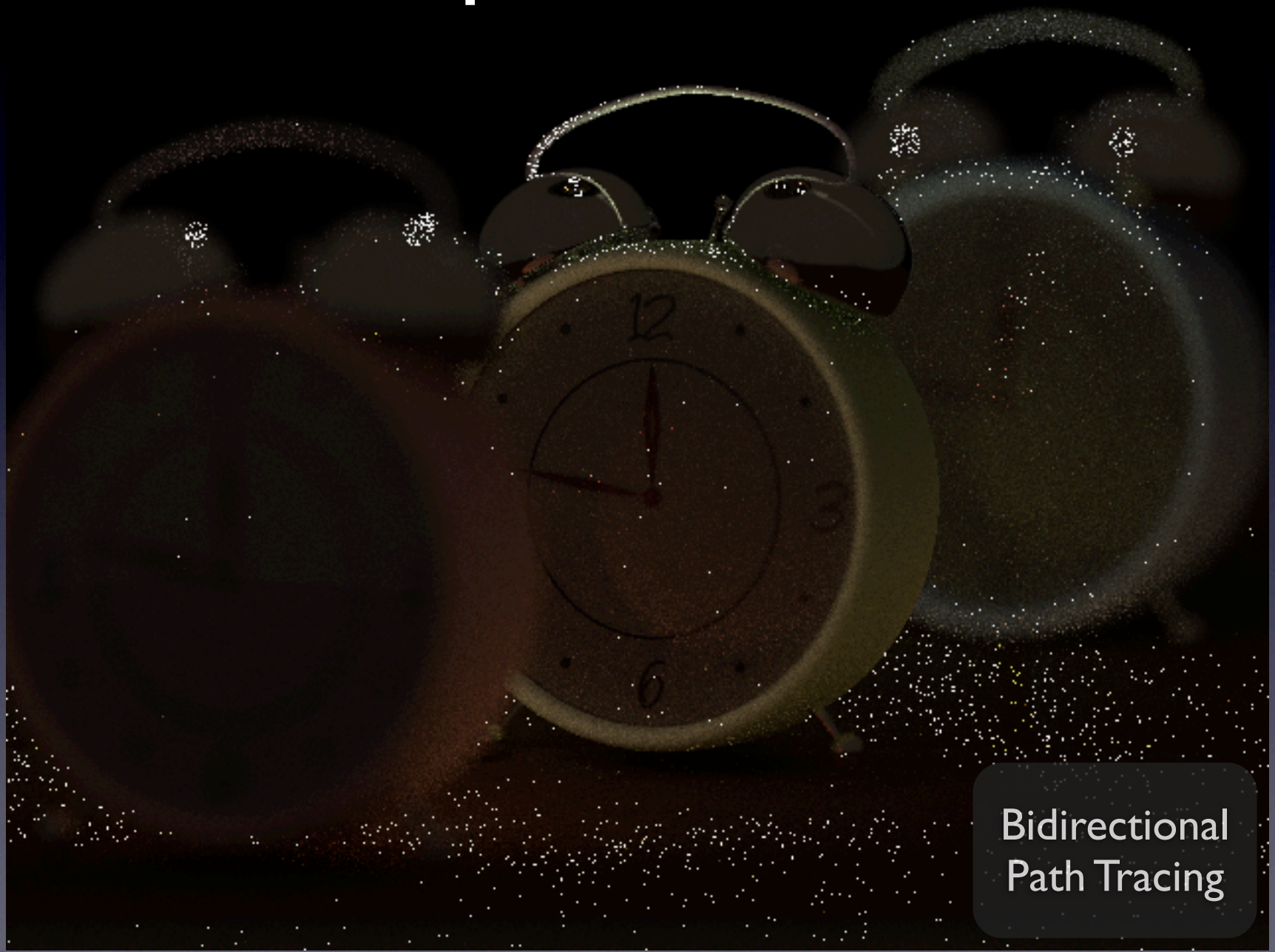
Convergence



Convergence



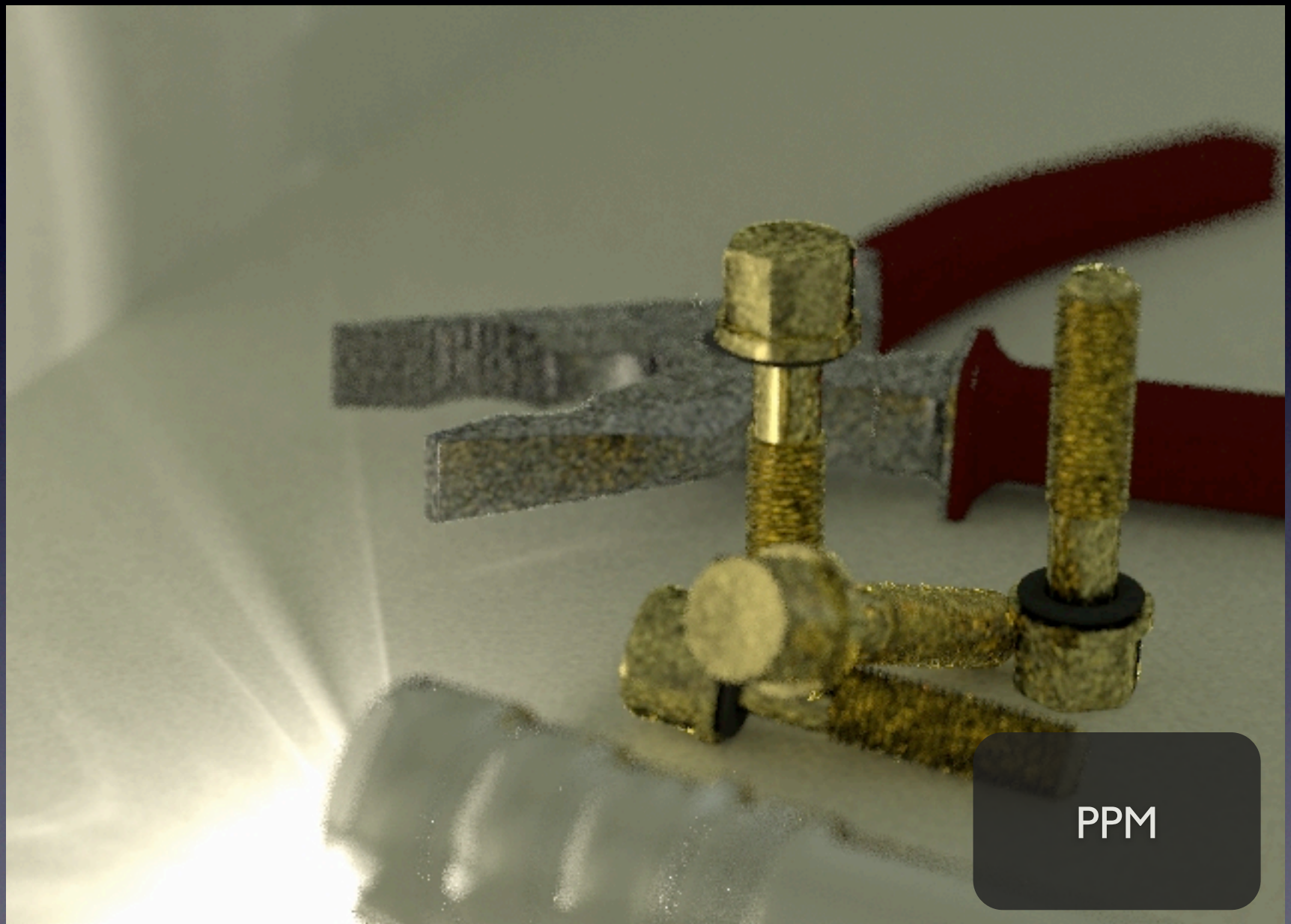
Depth-of-Field



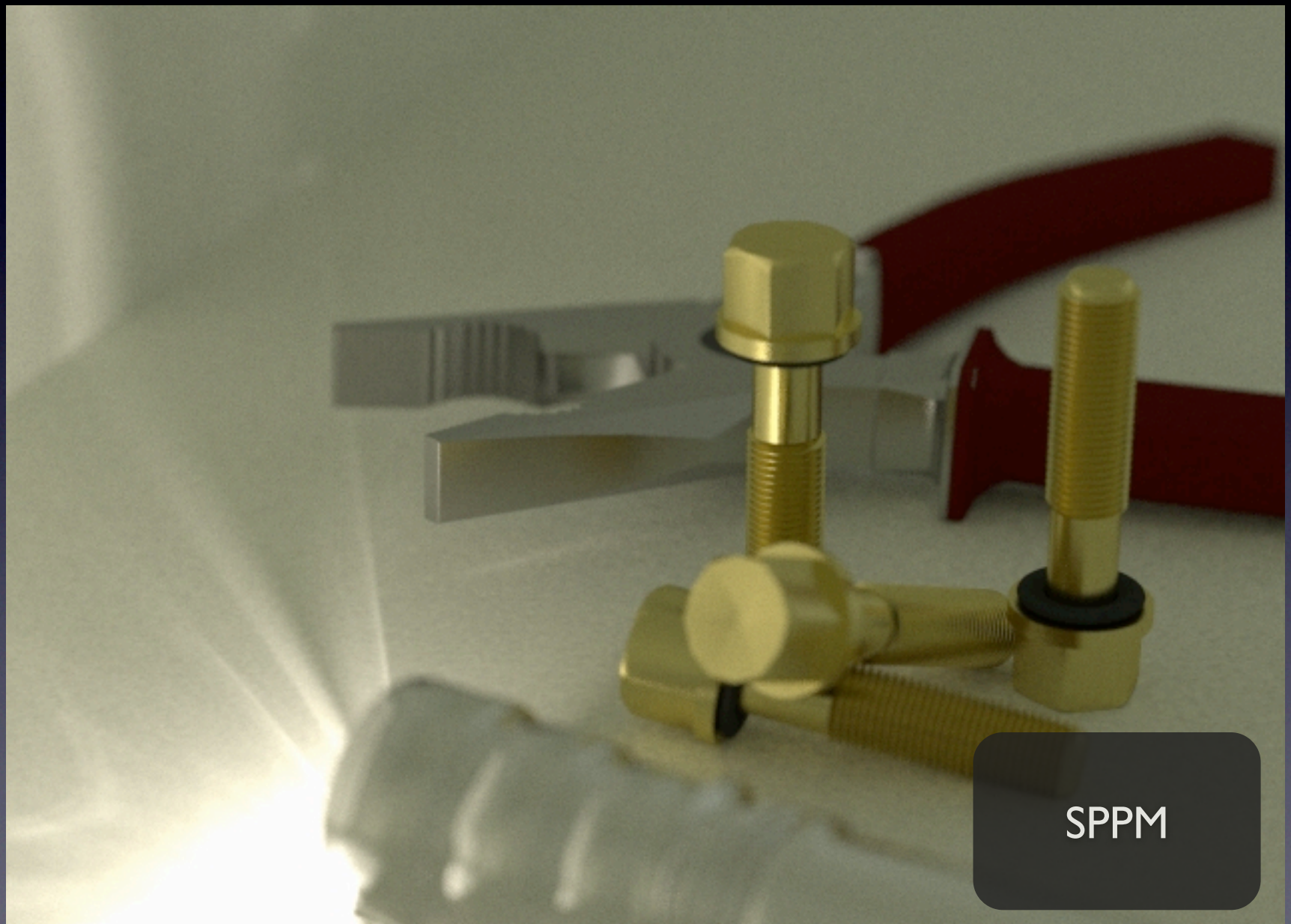
Depth-of-Field

Depth-of-Field

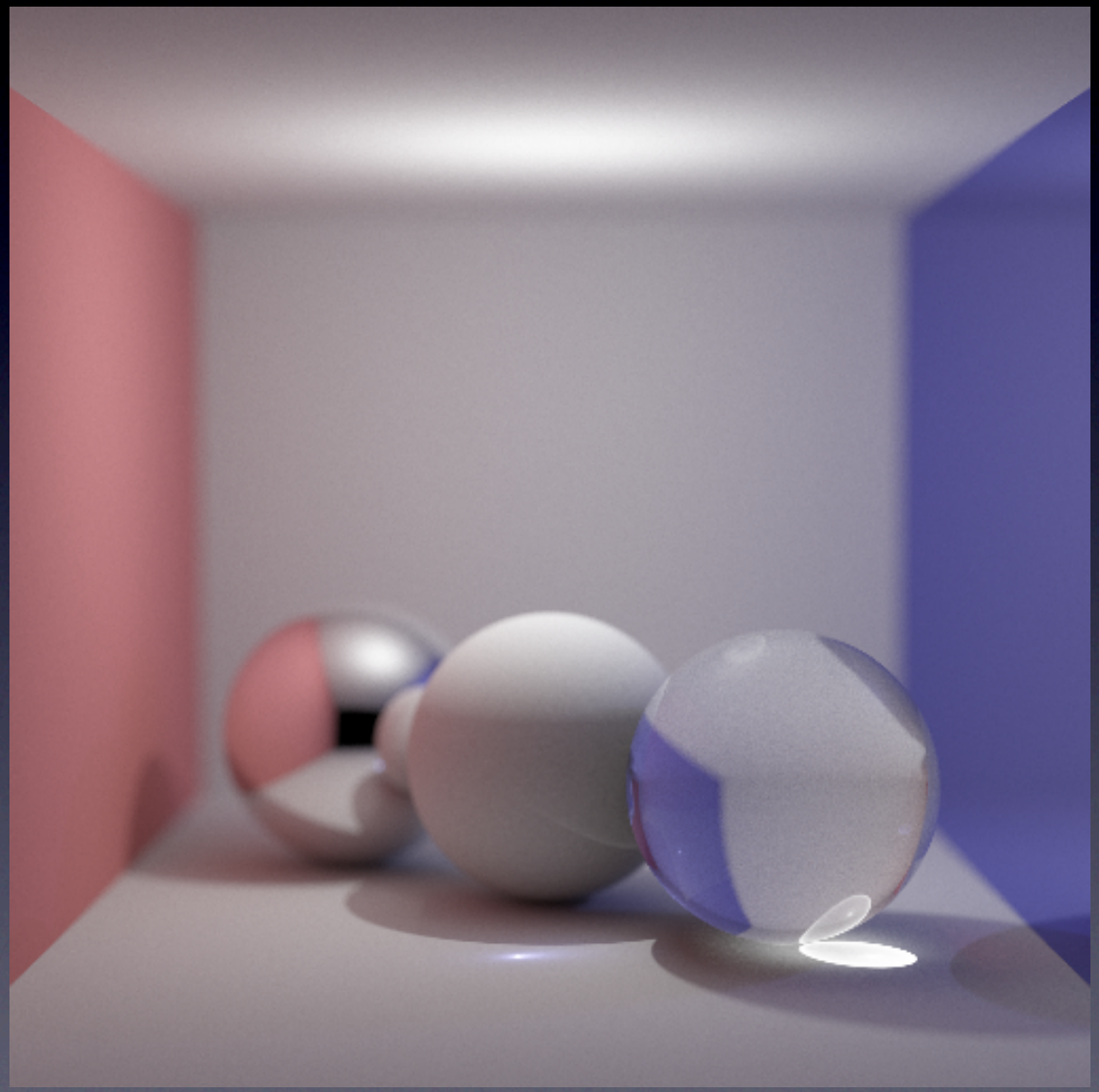
DOF + Glossy Reflection



DOF + Glossy Reflection



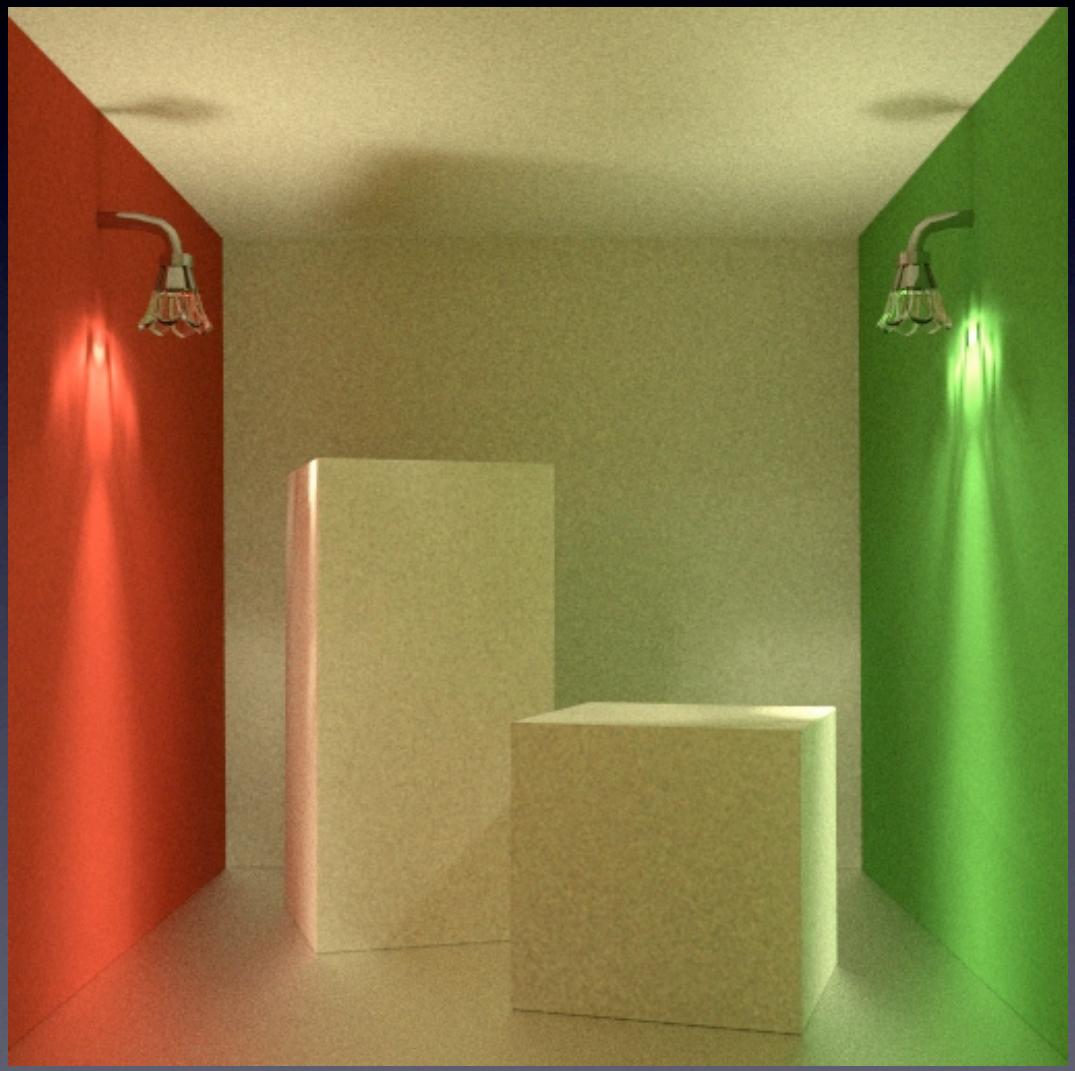
GPUSPPM



graphics.ucsd.edu/~toshiya

Conclusion

- Extension of progressive photon mapping
  - Consistent average radiance estimation
  - Can handle distributed ray tracing effects
  - Robust for *any* light path
  - Arbitrary accuracy using finite memory
  - Easy to implement

Acknowledgements

- ATI Fellowship 2008
- Youichi Kimura (flashlight, room)
- ShareCG.com (models)
- Will Chang, Wan-Yen Lo,
  Marios Papas, Iman Sadeghi (UCSD)
- Antoine Bouthors (Weta Digital)
- Wojciech Jarosz (Disney Research)
- Leonhard Grűnshloβ (NVIDIA)
- Anonymous reviewers