Progressive Photon Mapping Basics

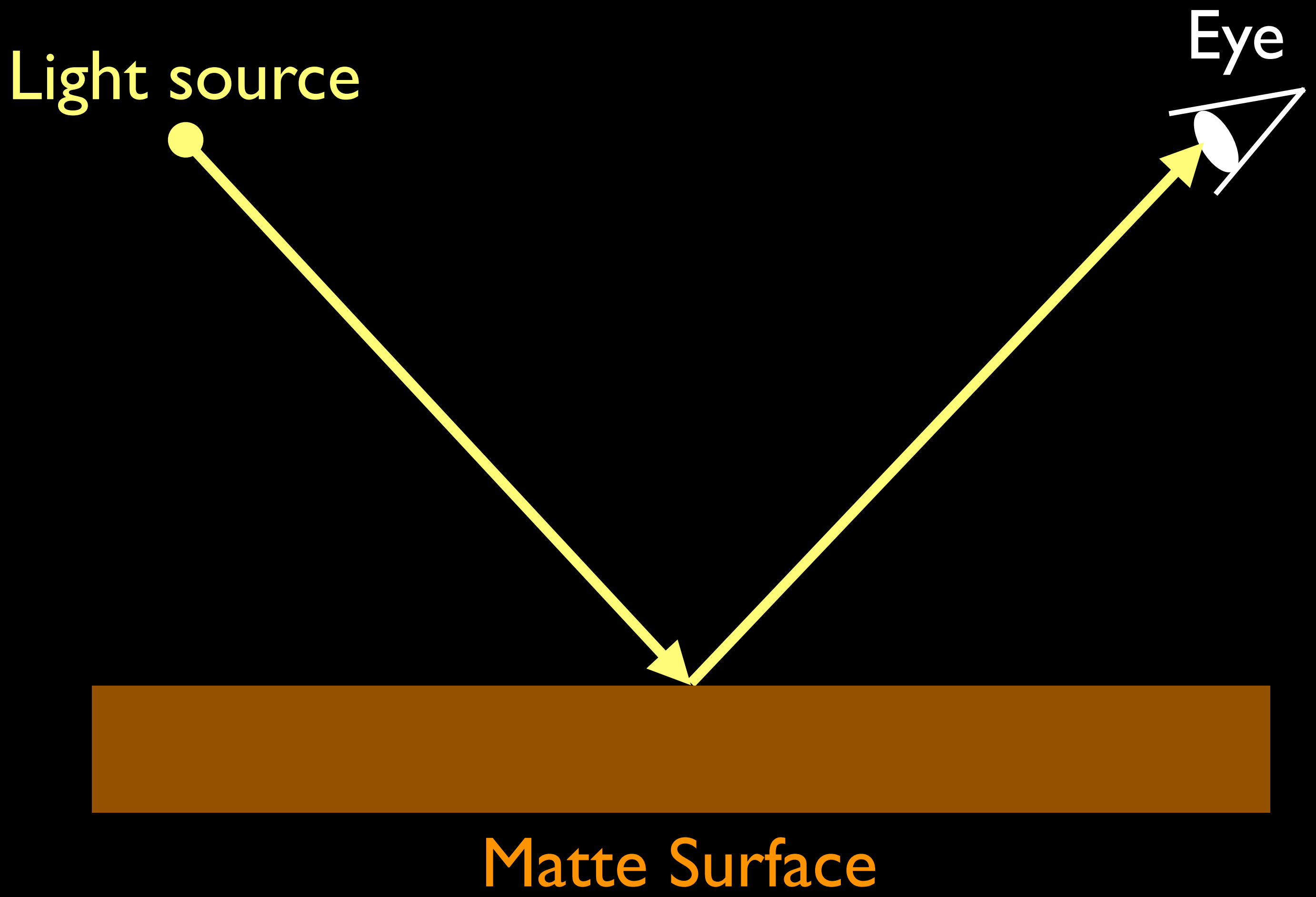
Toshiya Hachisuka
Aarhus University

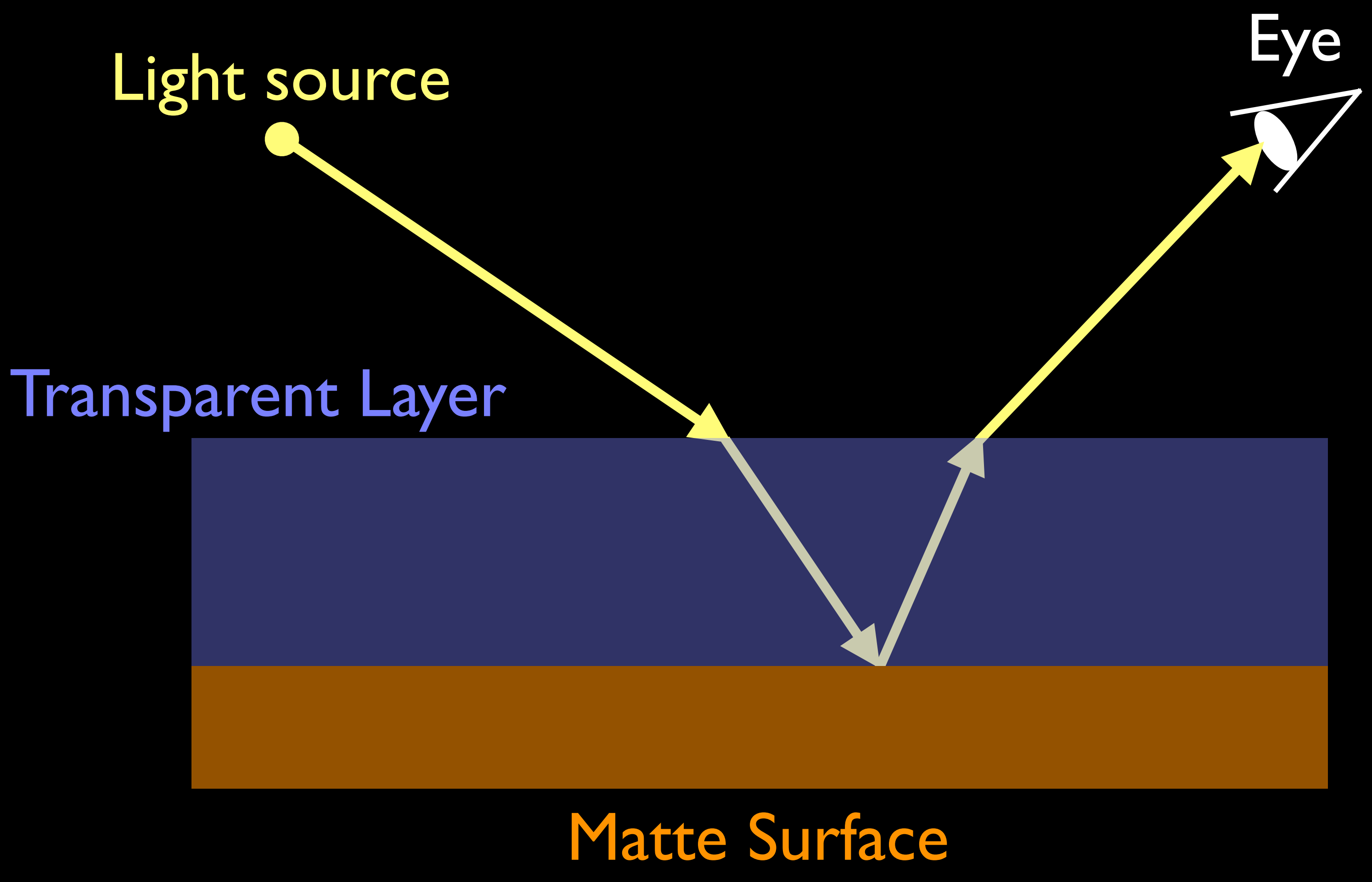
State of the Art in Photon Density Estimation

Light Transport Simulation Algorithms

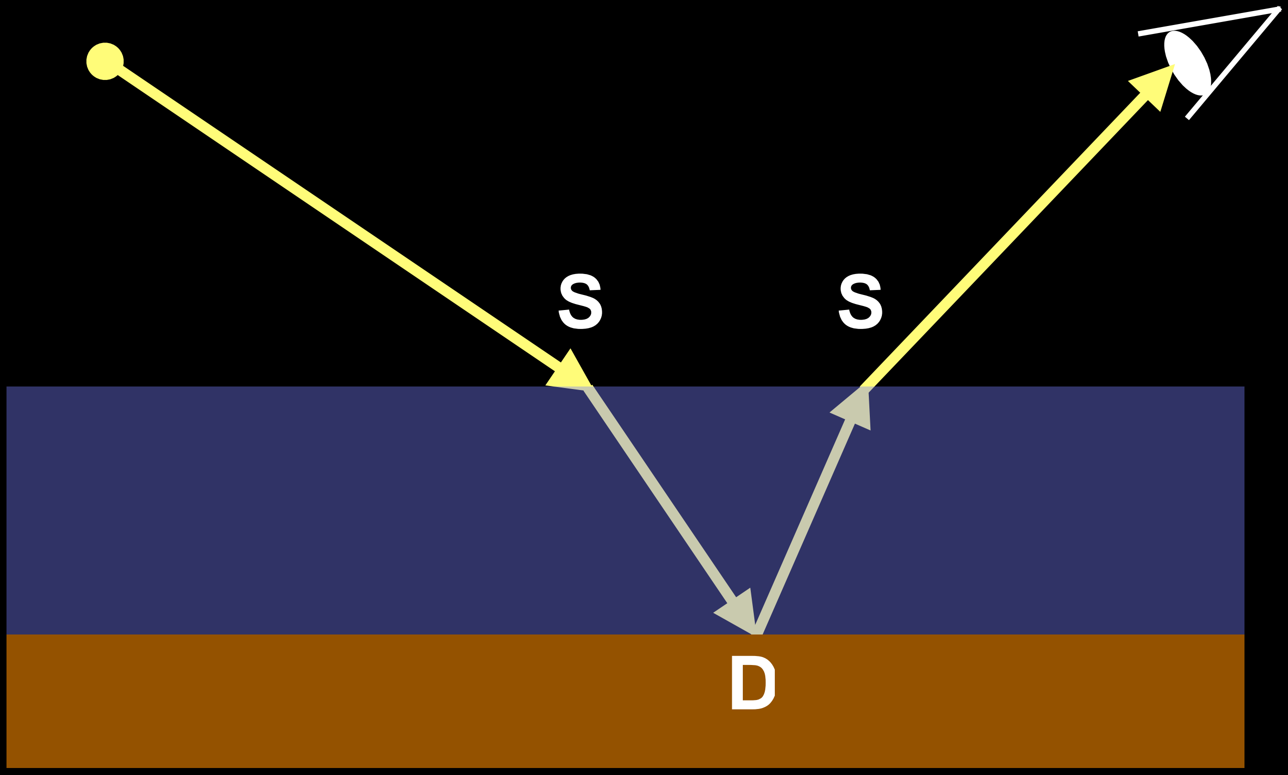
- Path Tracing [Kajiya 86]
- Light Tracing [Arvo 86][Dutré 93]
- Bidirectional Path Tracing [Lafortune 93][Veach 95]
- Photon Mapping [Jensen 95]
- Density Estimation [Shirley 95]
- Instant Radiosity [Keller 97]
- Metropolis Light Transport [Veach 97]
- Lightcuts [Walter 05]
- Energy Redistribution Path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

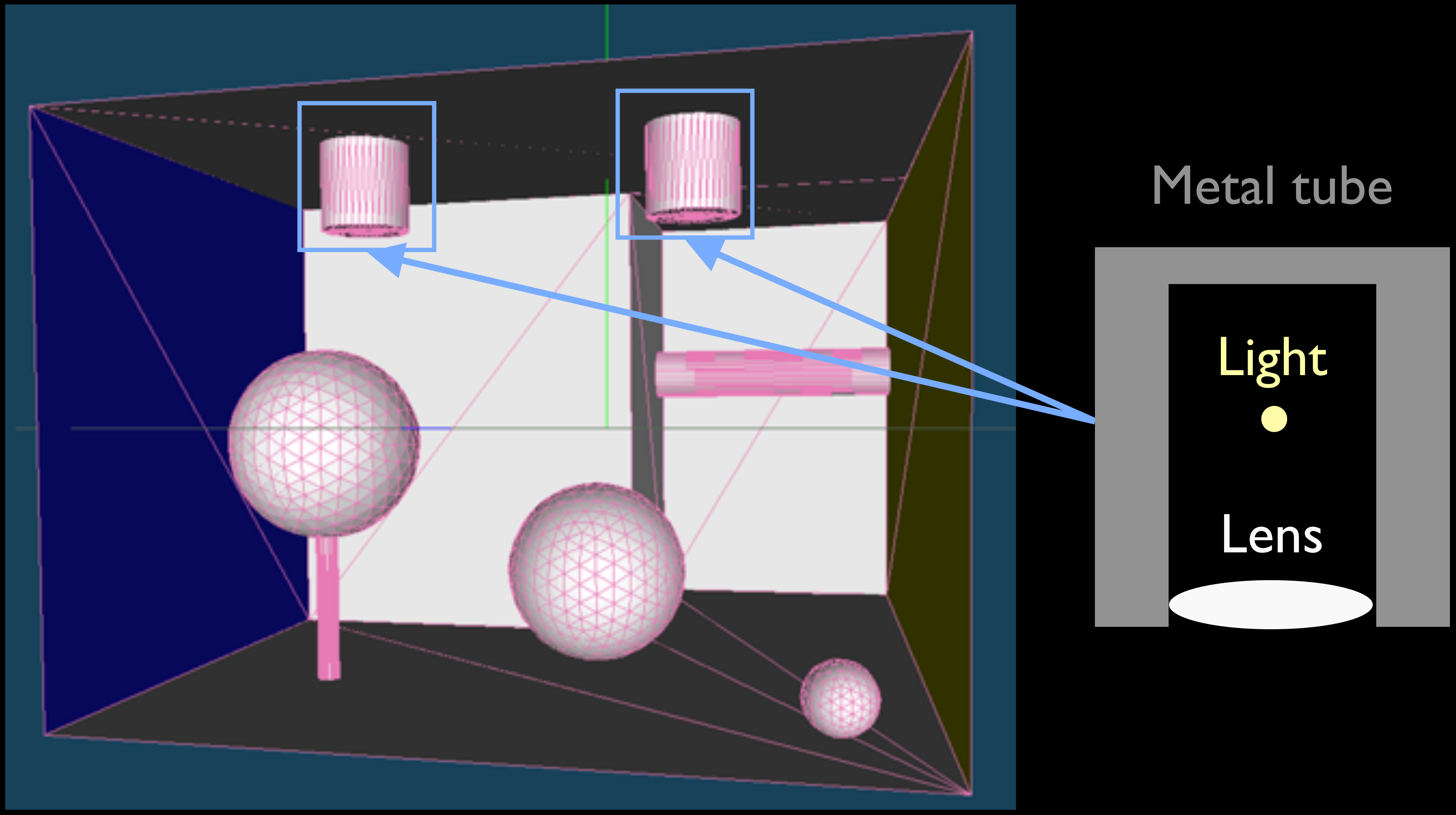

(c)Y. Kimura


Light source
Eye
Matte Surface

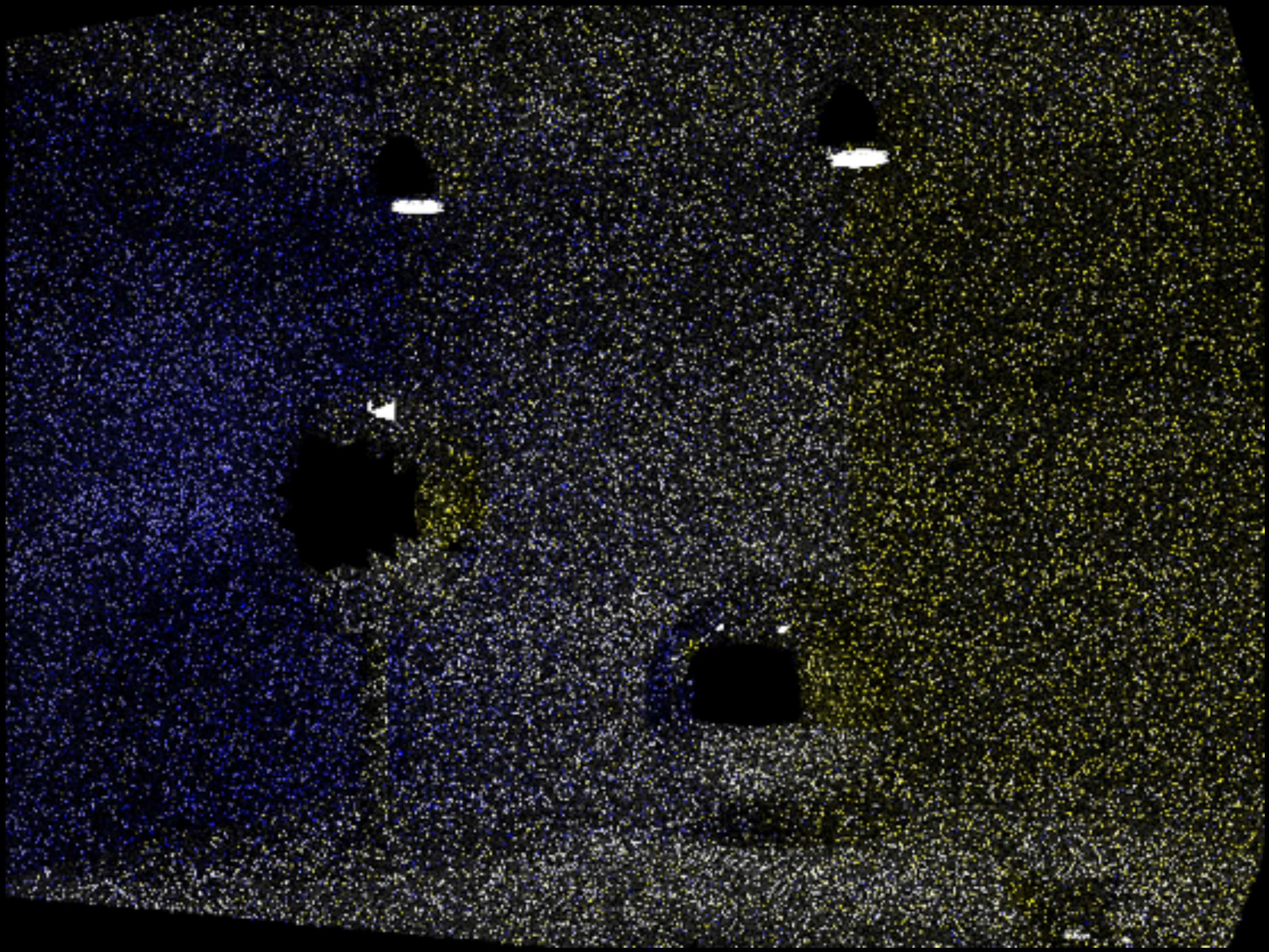

Eye
Light source
Transparent Layer
Matte Surface

Specular-Diffuse-Specular (SDS) paths

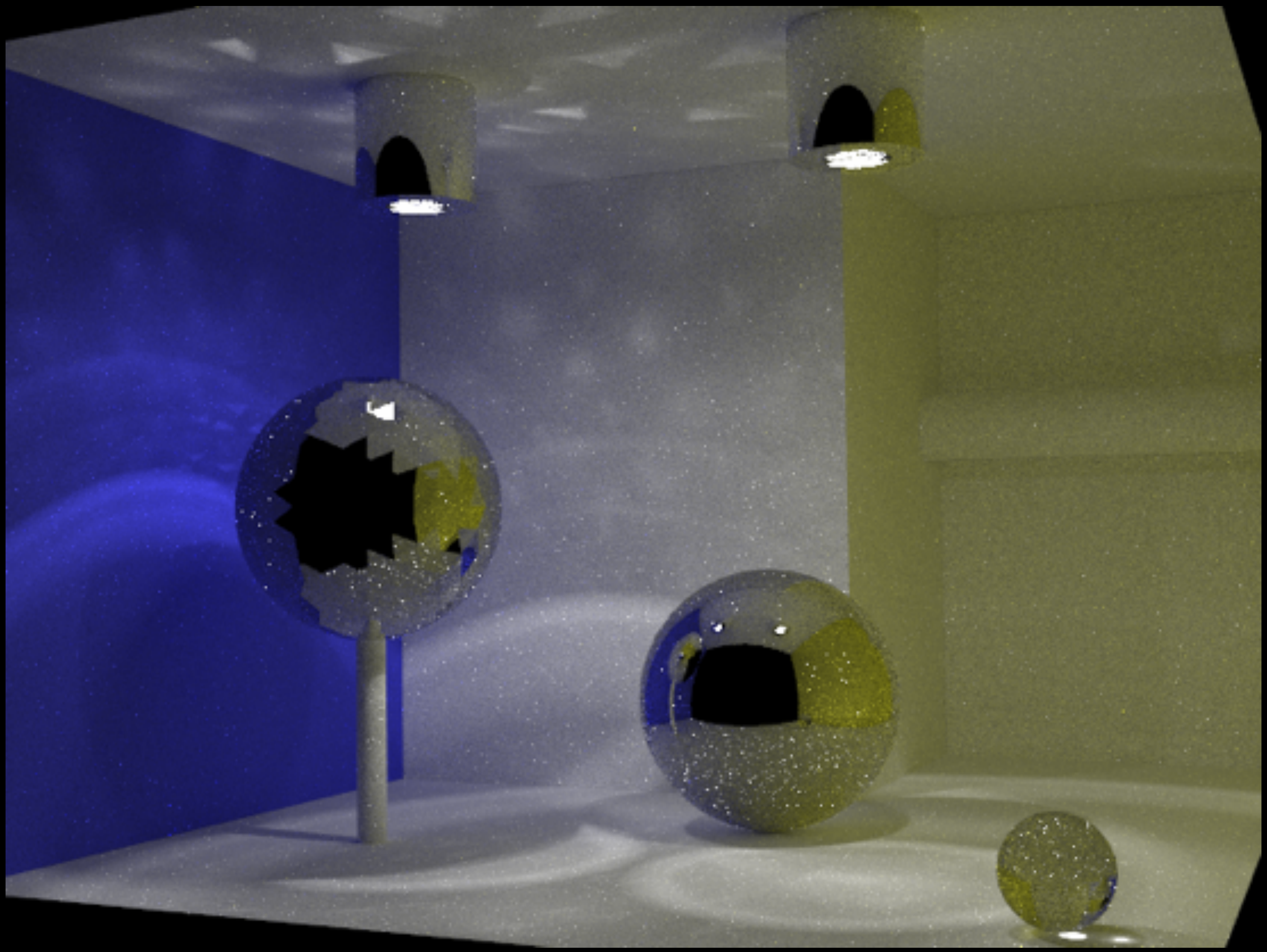



Metal tube
Light
Lens

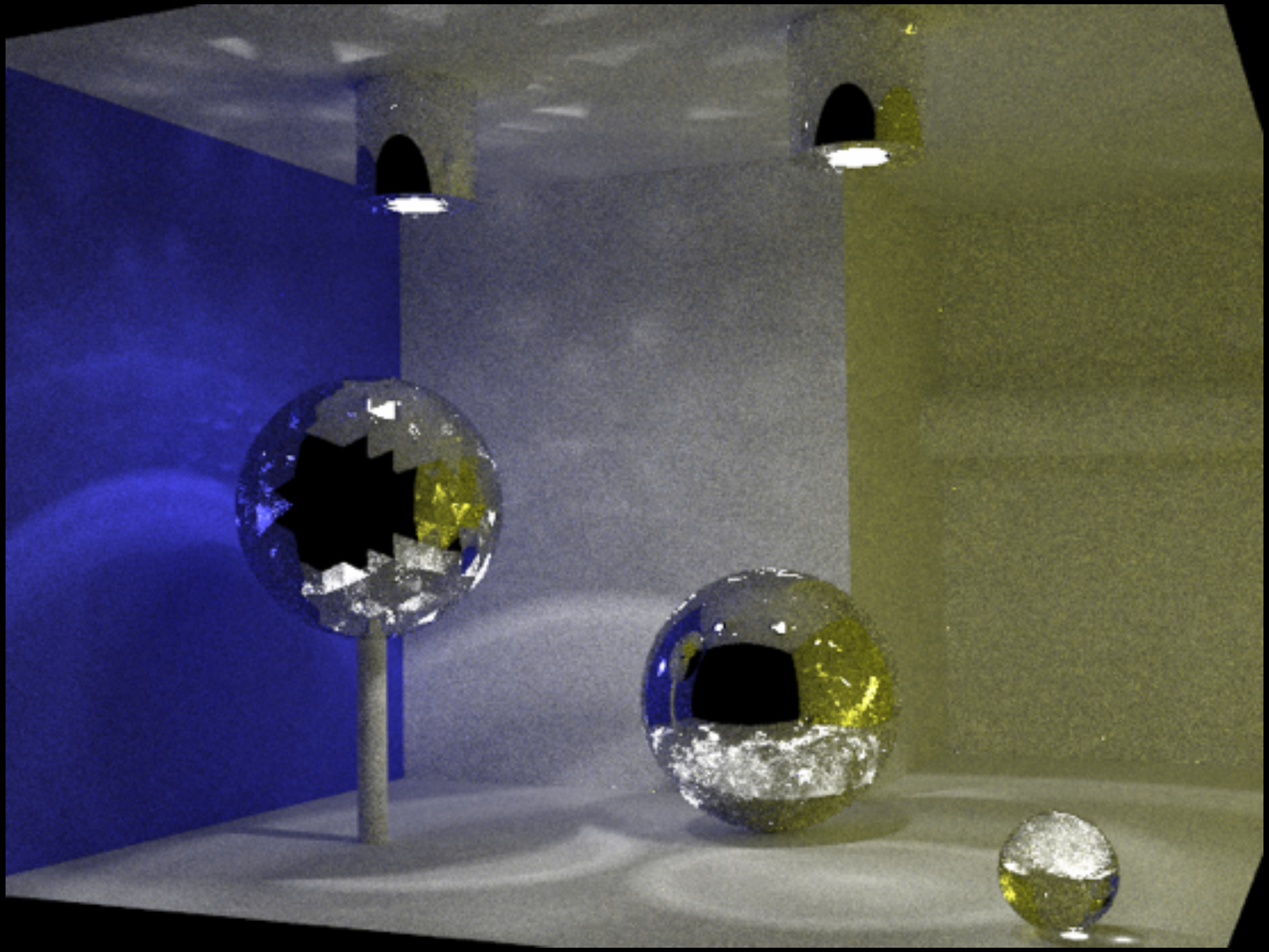
Path Tracing



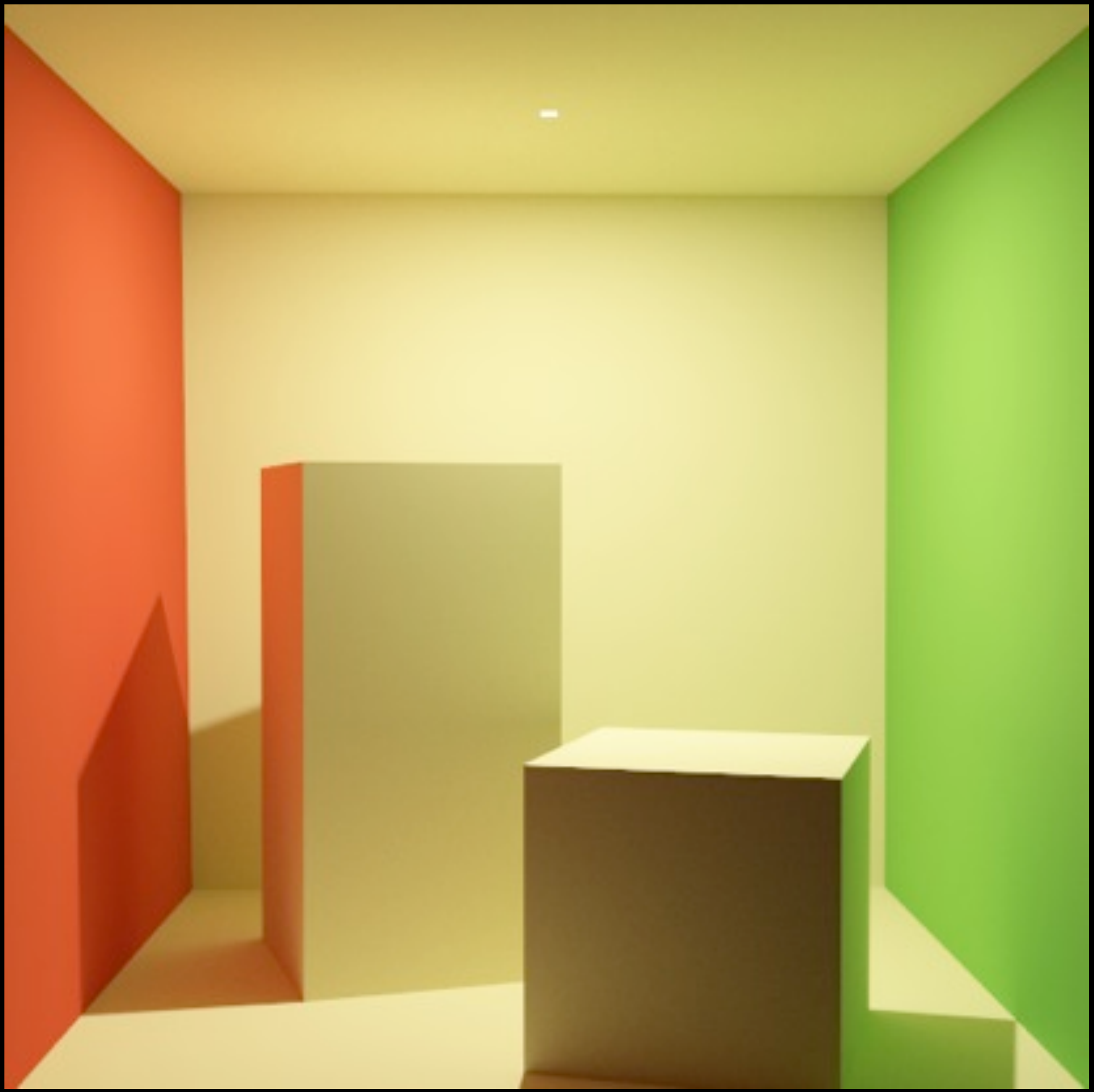
Bidirectional Path Tracing



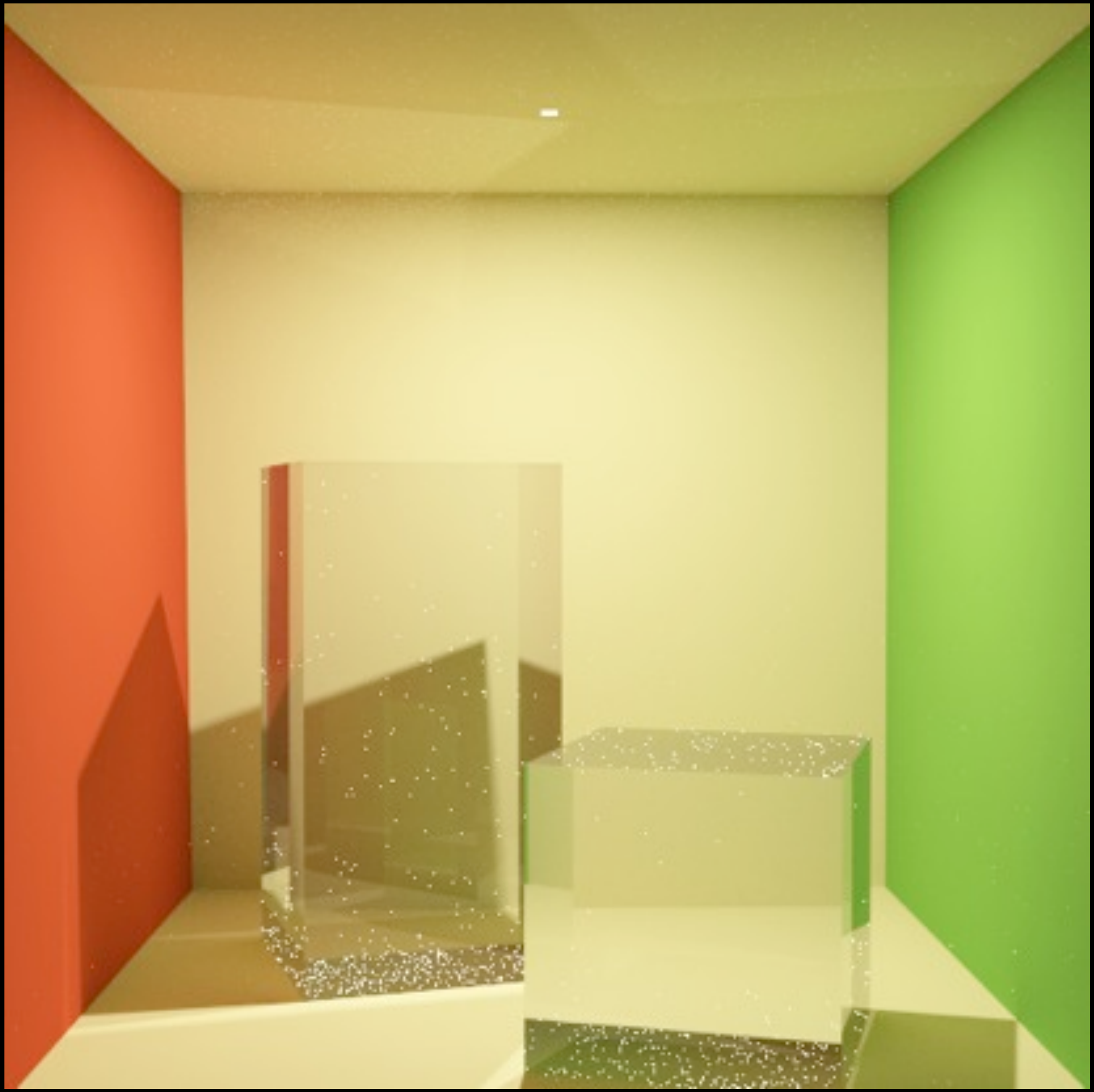
Metropolis Light Transport

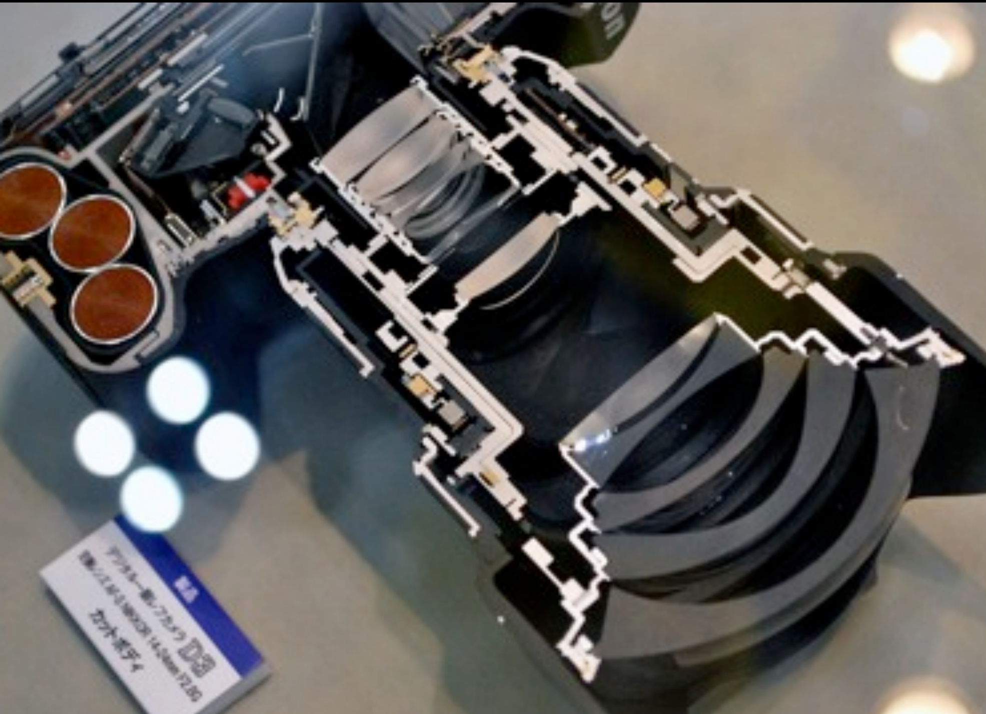


Bidirectional Path Tracing



Bidirectional Path Tracing





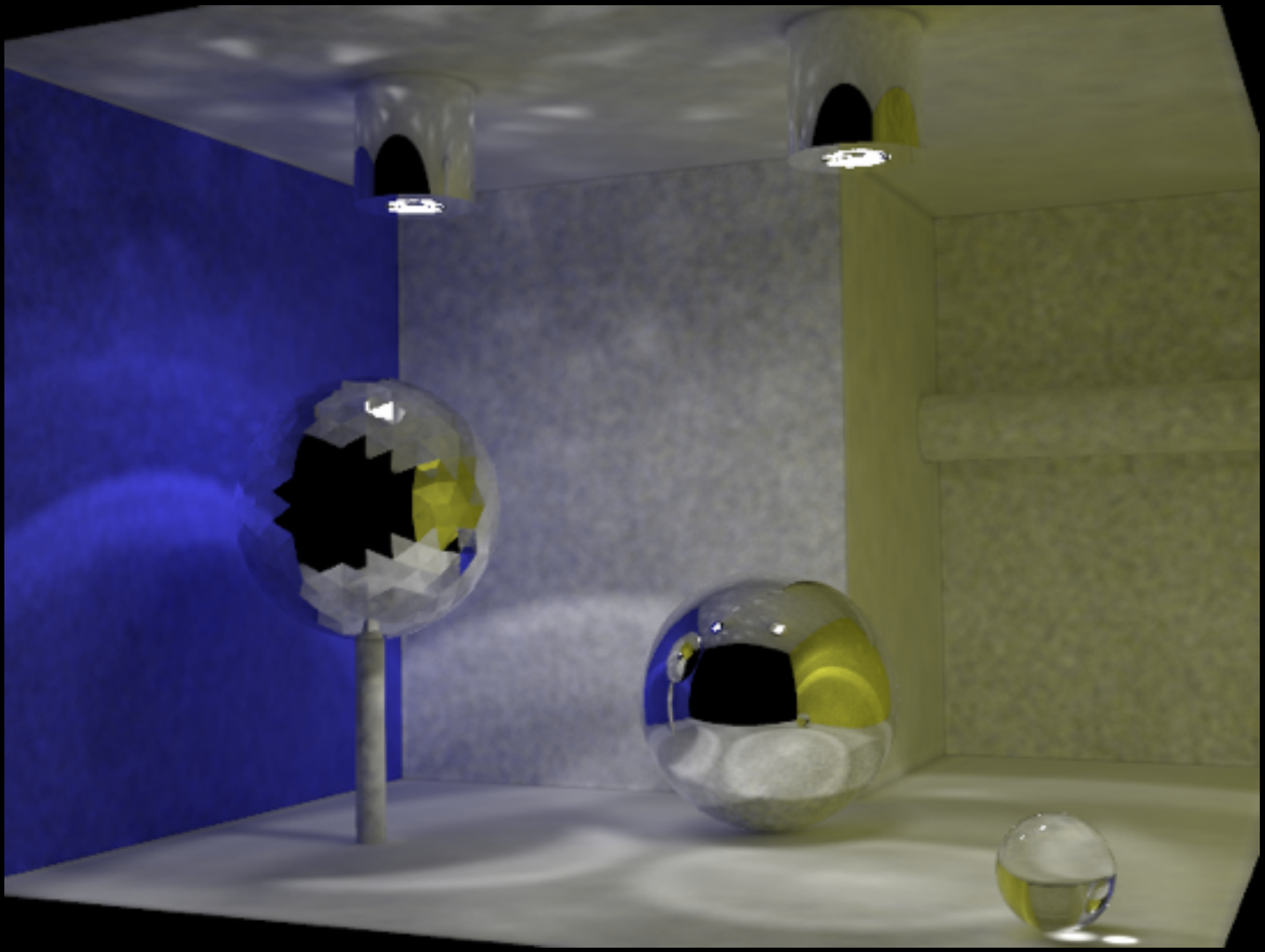

Specular-Diffuse-Specular Paths

- Existing methods are not robust for SDS paths
  - Path tracing
  - Bidirectional path tracing
  - Metropolis light transport
  - ...name your favorite

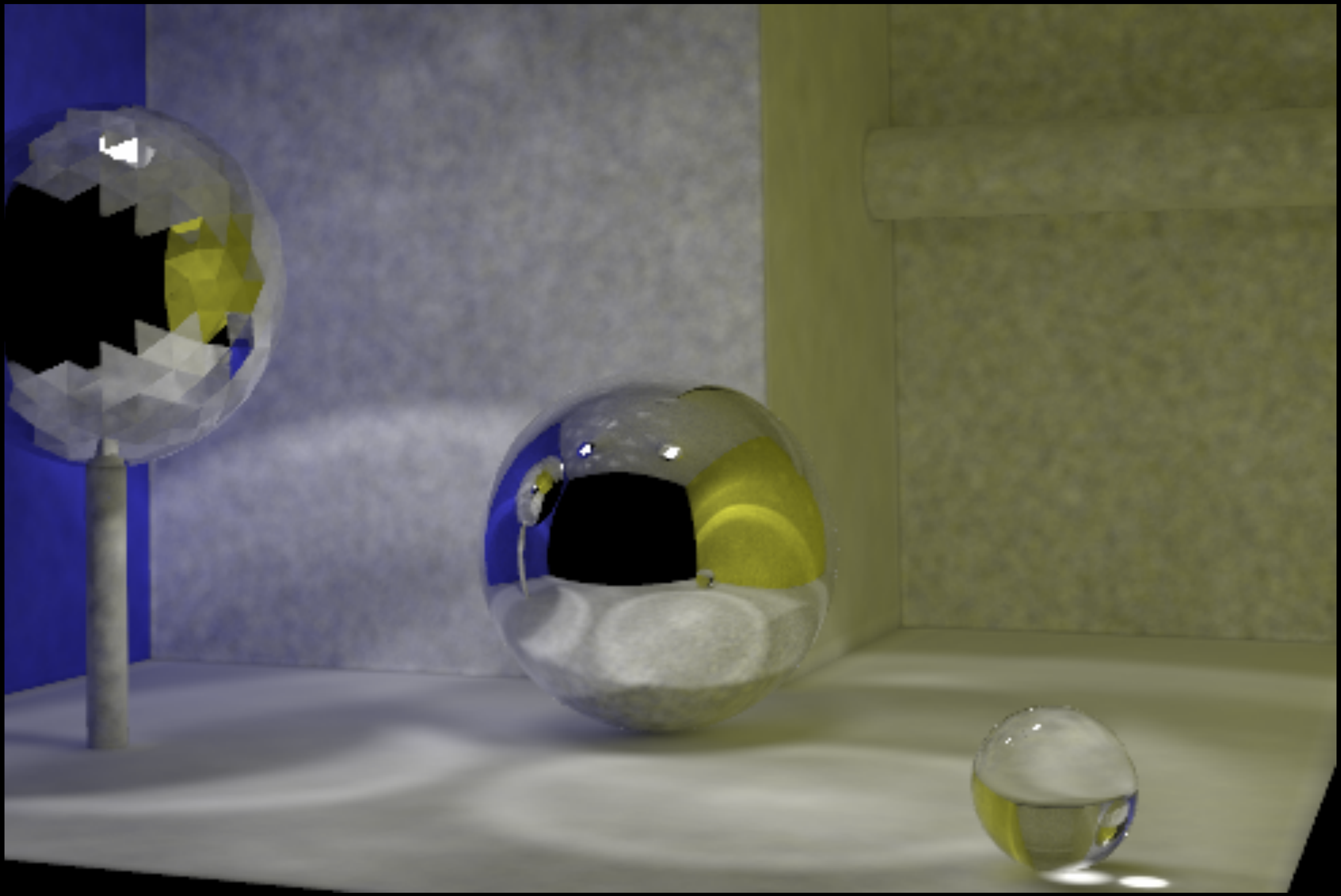
Specular-Diffuse-Specular Paths

- Existing methods are not robust for SDS paths
  - Path tracing
  - Bidirectional path tracing
  - Metropolis light transport
  - ...name your favorite
  - Photon mapping?

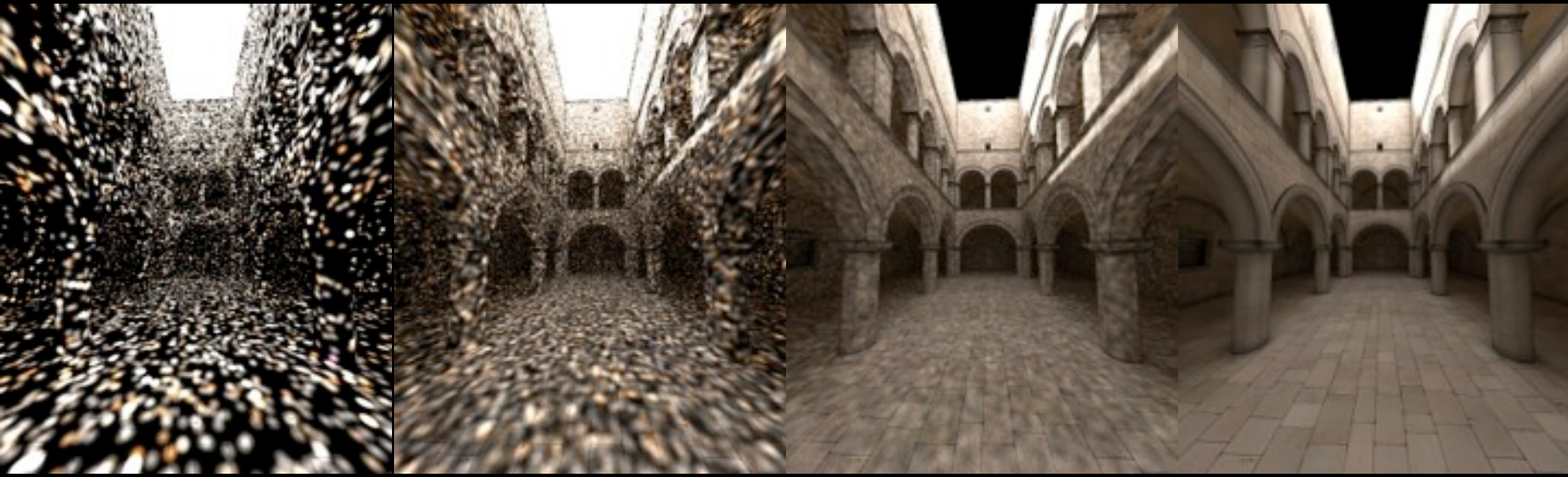
Photon Mapping



Photon Mapping



Convergence of Photon Mapping



More photons →

Convergence of Photon Mapping

$$L(x,\vec{\omega}) = \lim_{N\to\infty} \sum_{p=1}^{N^\beta} \frac{f_r(x,\vec{\omega},\vec{\omega}_p)\phi_p(x_p,\vec{\omega}_p)}{\pi r^2}$$

- Infinite number of nearby photons ($N^\beta \to \infty$)
- Infinitely small radius ($r \to 0$)

Convergence of Photon Mapping

$$L(x,\vec{\omega}) = \lim_{N\to\infty} \sum_{p=1}^{N^{\beta}} \frac{f_r(x,\vec{\omega},\vec{\omega}_p)\phi_p(x_p,\vec{\omega}_p)}{\pi r^2}$$

Infinite storage & photon tracing

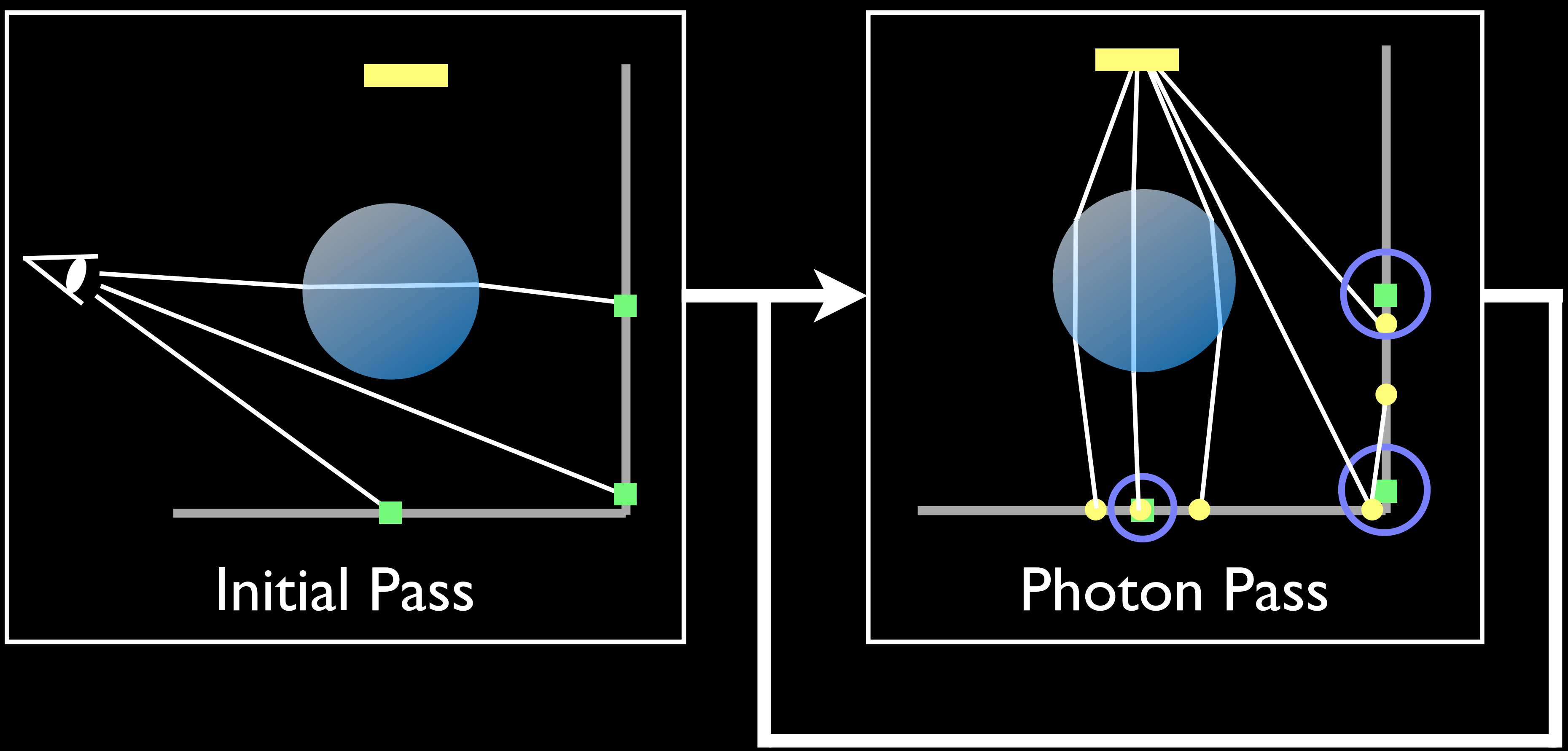
- Infinite number of nearby photons ($N^{\beta}\to\infty$)
- Infinitely small radius ($r\to 0$)

Convergence of Photon Mapping

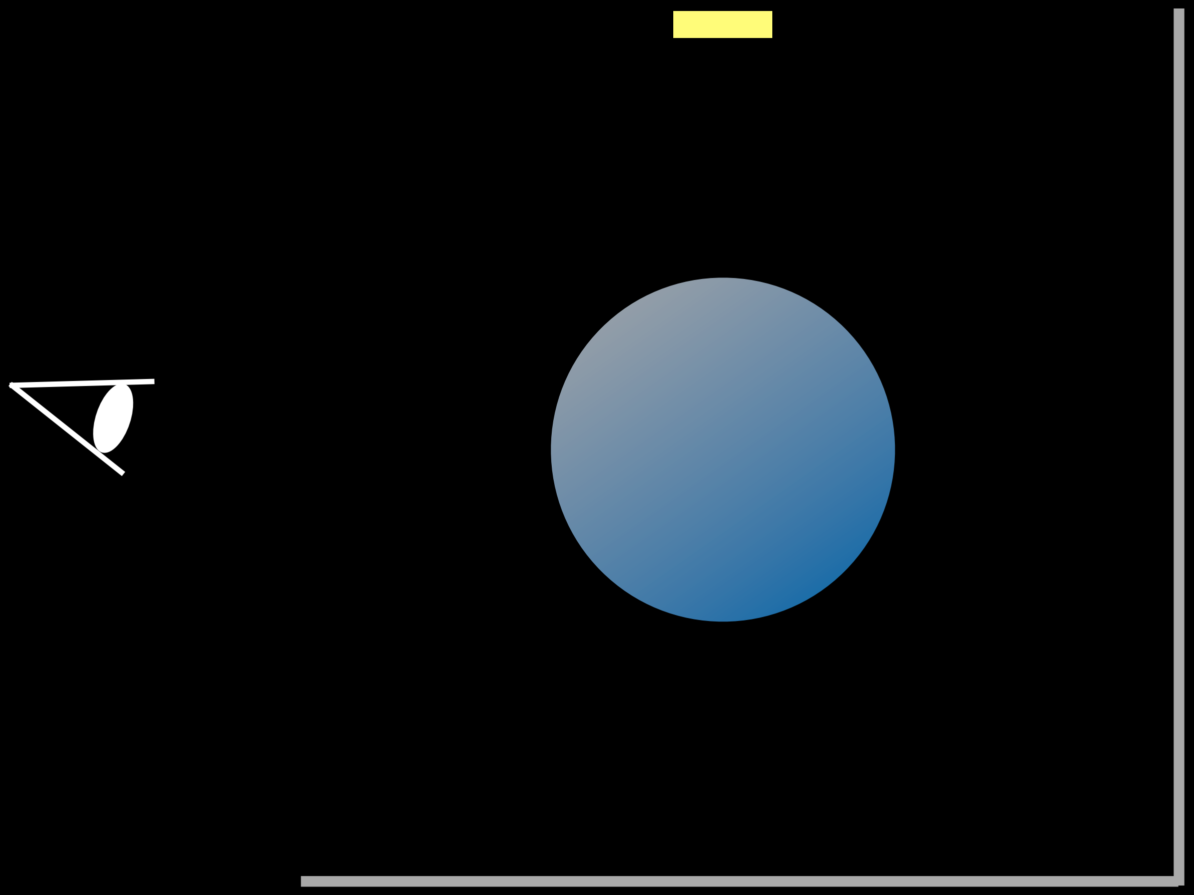
- Conditions of convergence
  - Infinite number of (neighboring) photons
  - Infinitely small radius
- In practice...
  - More memory
  - Longer waiting time to the image
  - Rerun the process with different parameters

Solution:
Progressive Photon Mapping

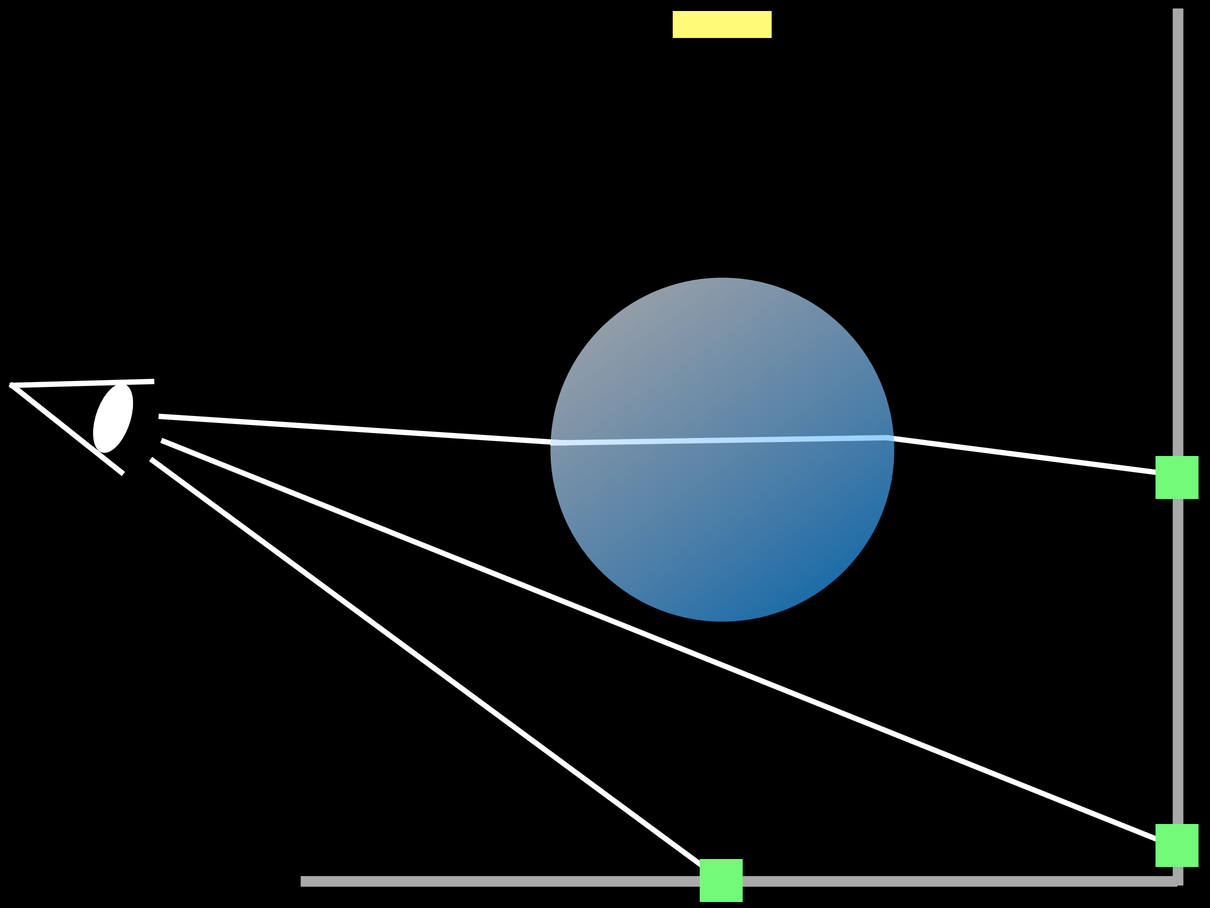
Overview



Scene

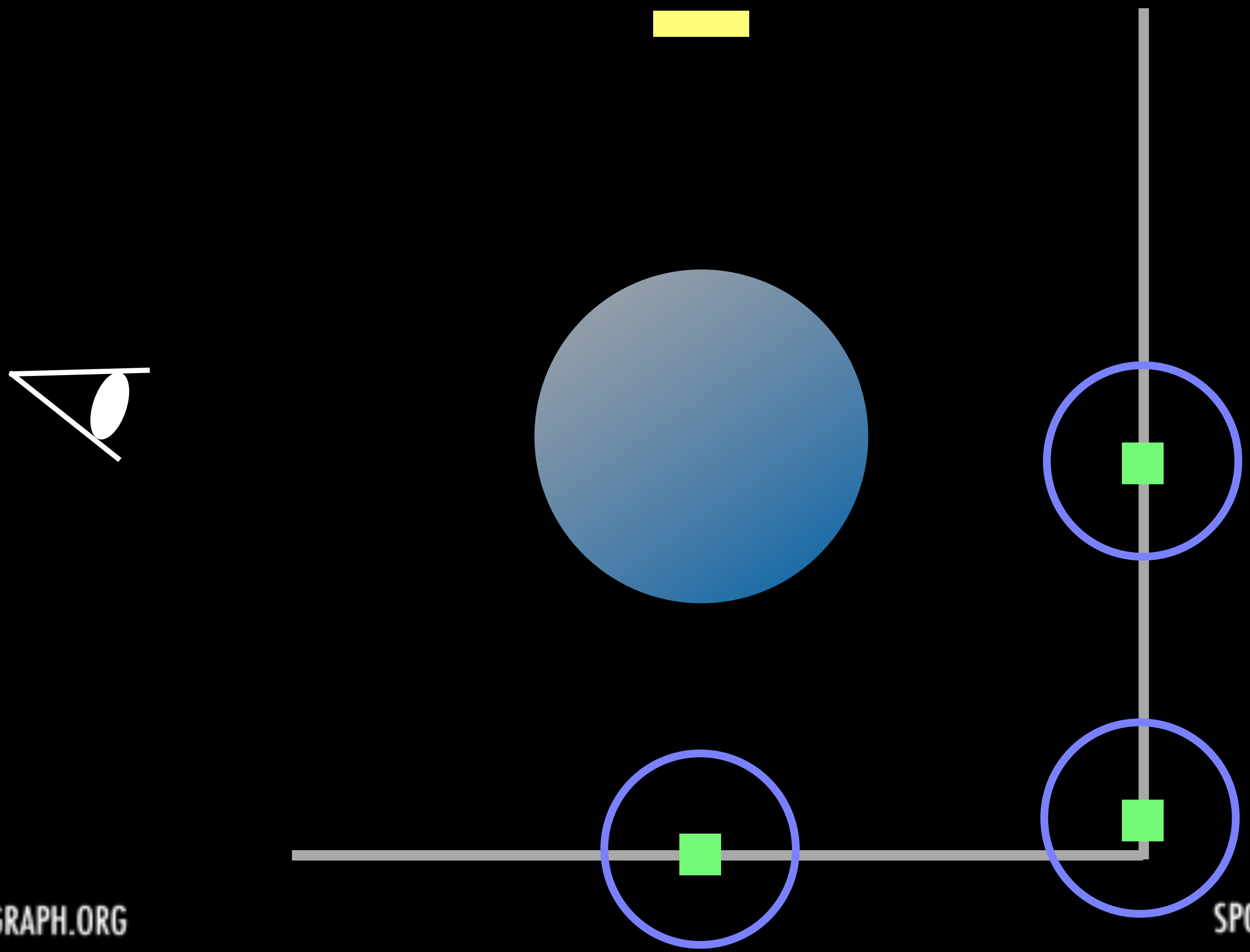


Initial Pass



Initial Pass

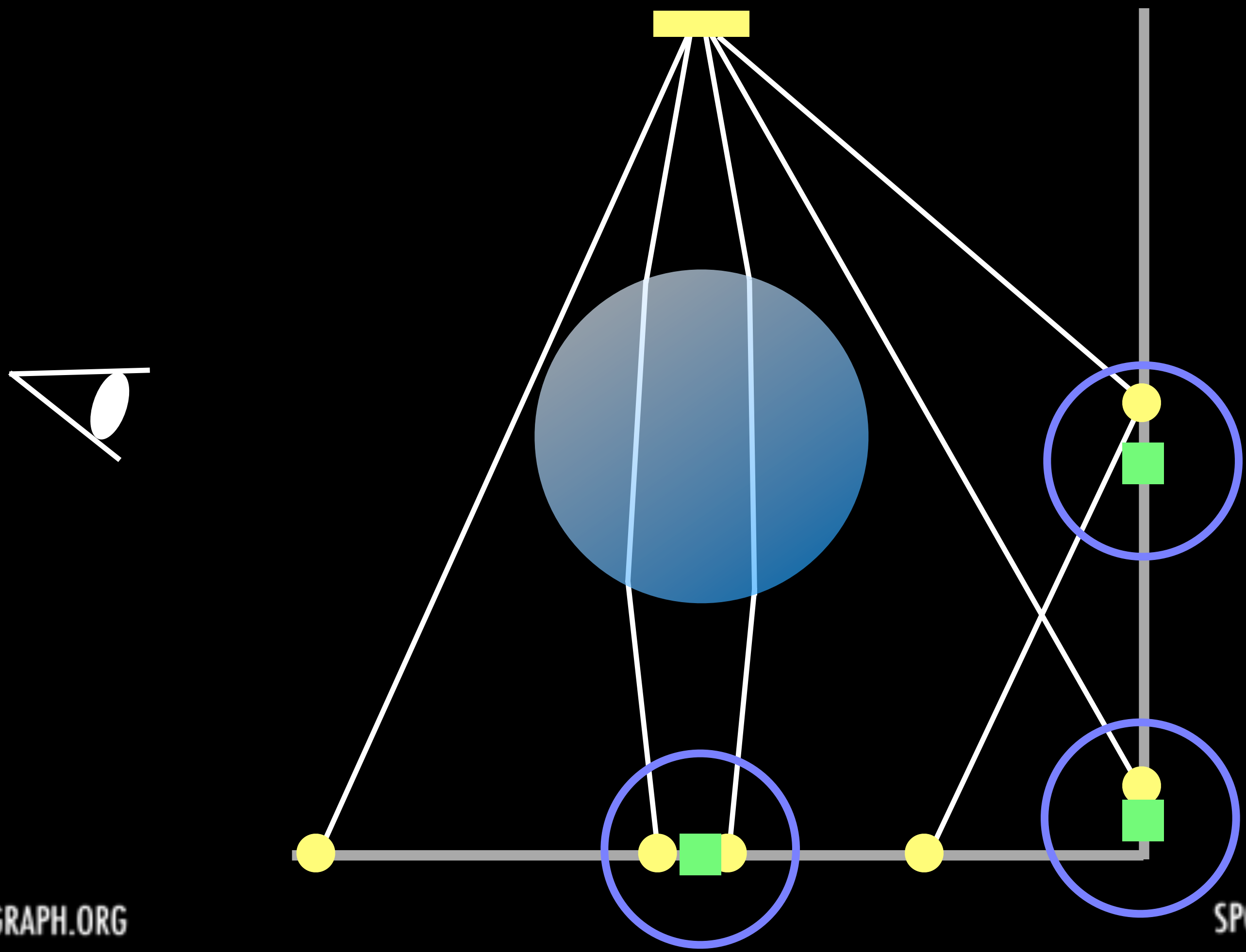
* Similar to Havran et al. 2008



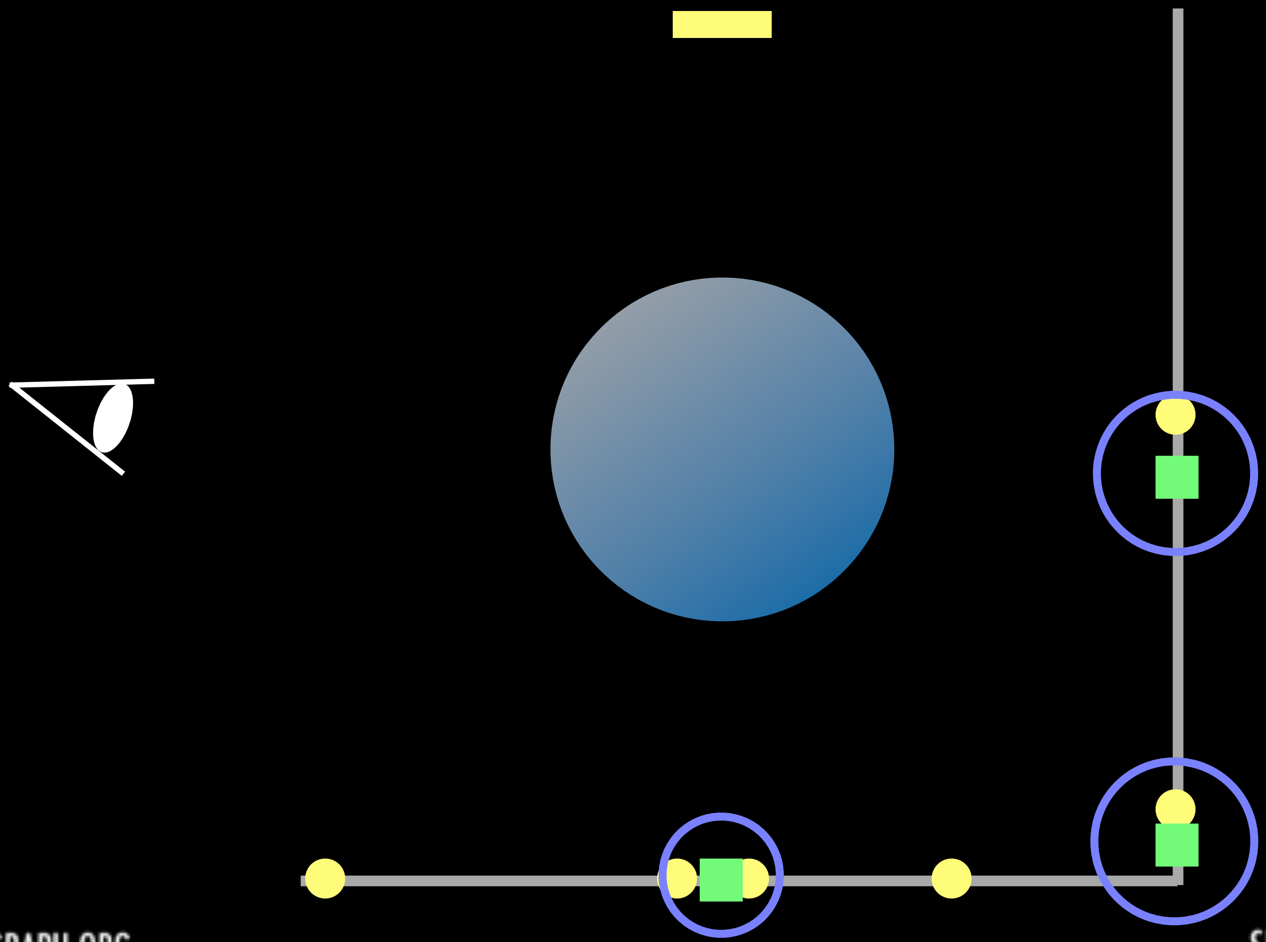
Photon Statistics

- Each measurement 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 $N_e(i)$

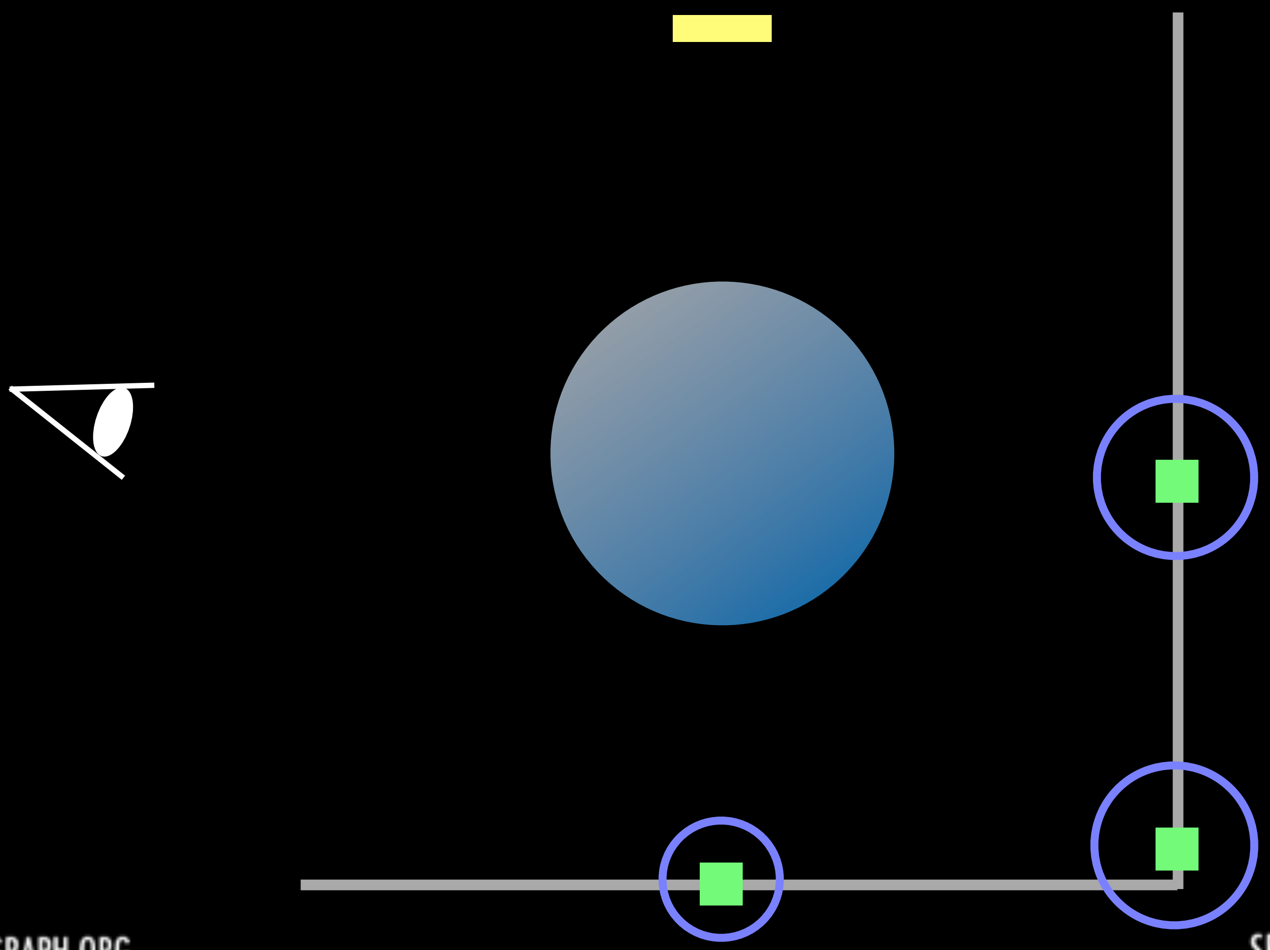
Photon Pass



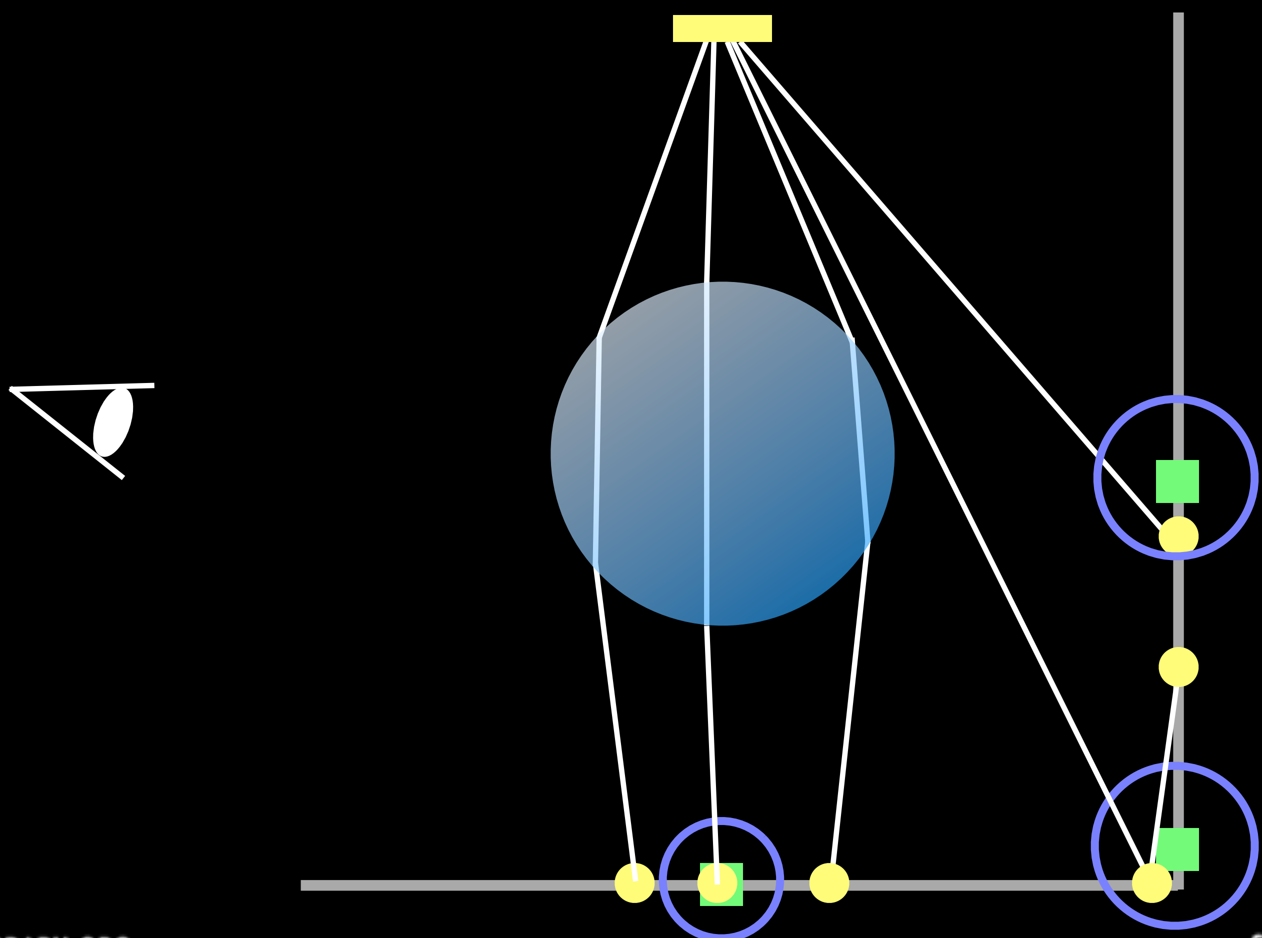
Photon Pass



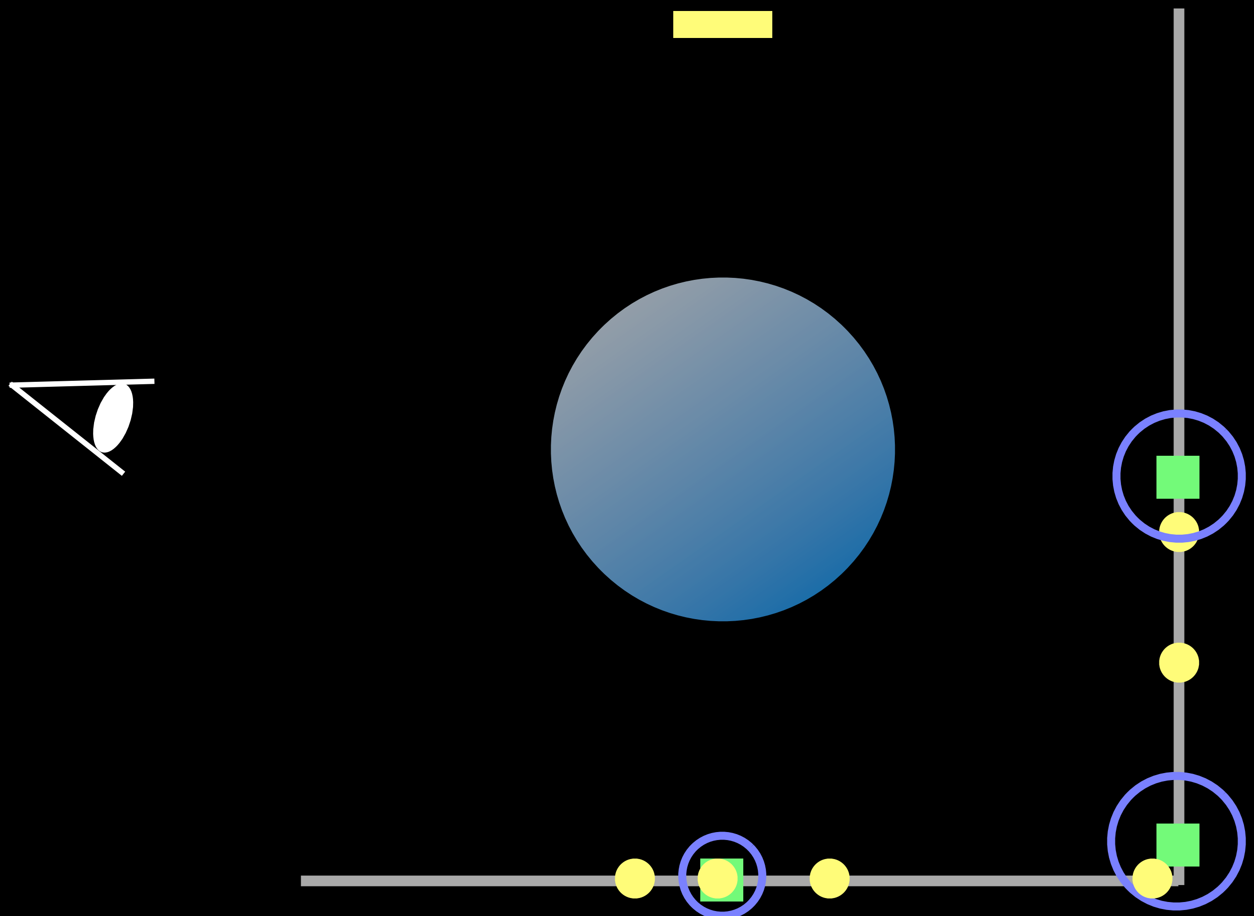
Photon Pass



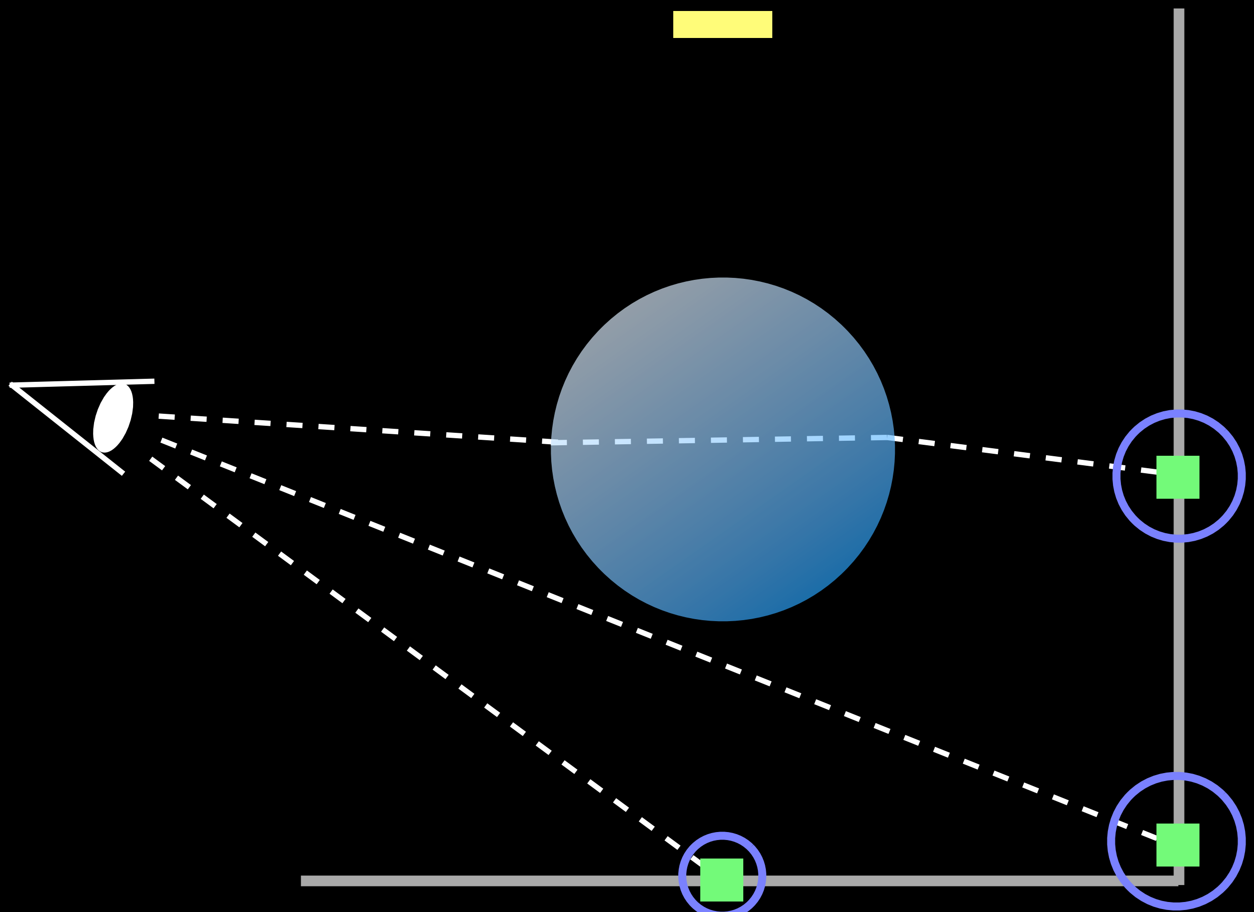
Next Photon Pass



Next Photon Pass



Rendering



Progressive Density Estimation

$$L_i(\vec{\omega}) = \sum_{p=1}^{N_i} \frac{f_r(\vec{\omega}, \vec{\omega}_p)\, \phi_p(\vec{\omega}_p)}{\pi R_i^2}$$

- Converges to the correct solution
  - Infinite number of photons
  - Infinitely small radius

Progressive Density Estimation

$$L_i\left(\vec{\omega}\right) = \sum_{p=1}^{N_i} \frac{f_r\left(\vec{\omega}, \vec{\omega}_p\right)\phi_p\left(\vec{\omega}_p\right)}{\pi R_i^2}$$

- Converges to the correct solution
  - $\lim_{i \to \infty} N_i(R_i) = \infty$
  - $\lim_{i \to \infty} R_i(N_i) = 0$

Progressive Density Estimation

SIGGRAPHASIA2013

$$L_i(\vec{\omega}) = \sum_{p=1}^{N_i} \frac{f_r(\vec{\omega}, \vec{\omega}_p)\, \phi_p(\vec{\omega}_p)}{\pi R_i^2}$$

- Converges to the correct solution
  - $\lim_{i \to \infty} N_i(R_i) = \infty$
  - $\lim_{i \to \infty} R_i(N_i) = 0$

Recursive density estimation using sample statistics

Progressive Density Estimation

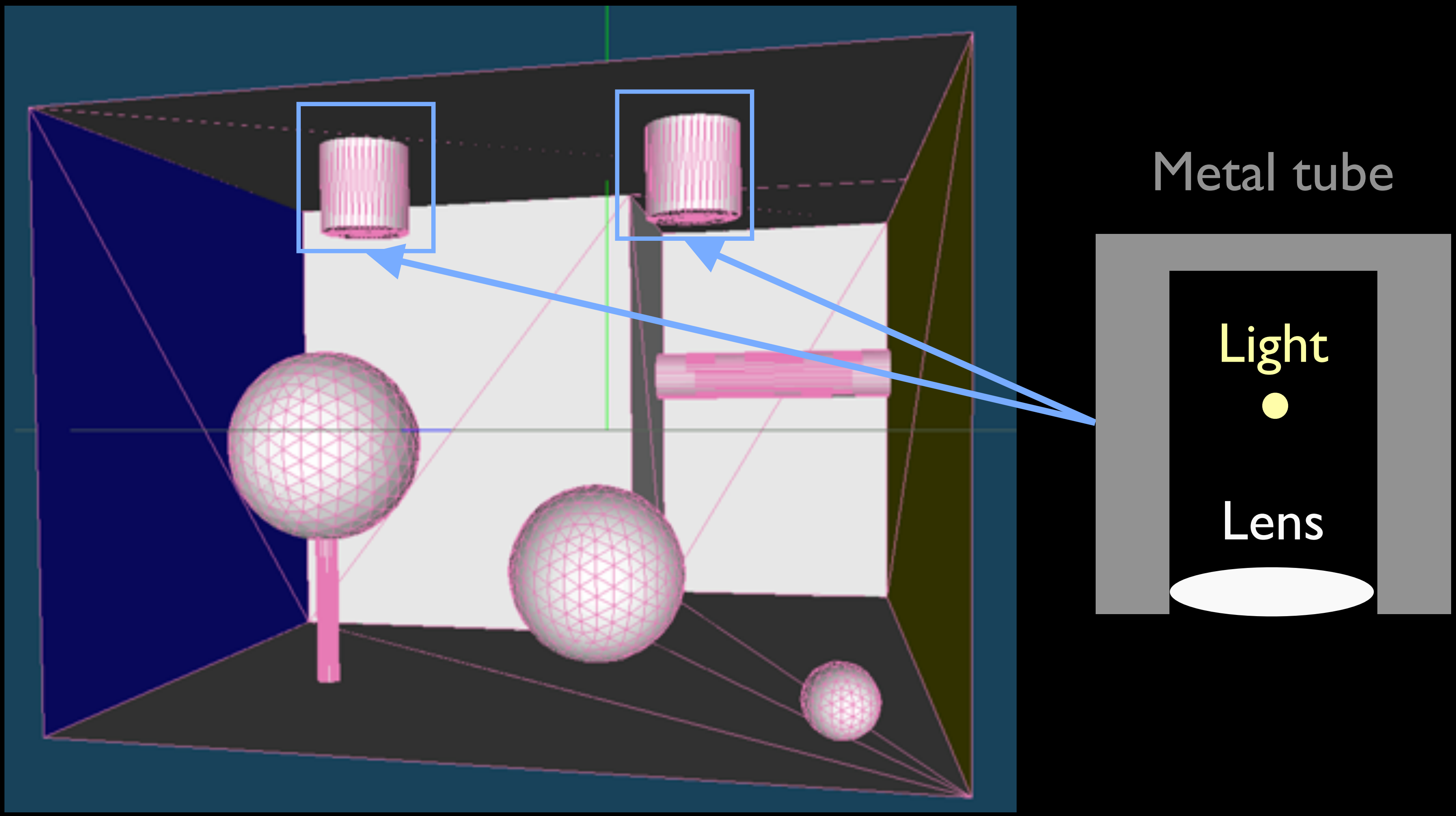
Number of photons $$N_{i+1} = N_i + \alpha M_i$$

Radius $$R_{i+1} = R_i \sqrt{\frac{N_i + \alpha M_i}{N_i + M_i}}$$

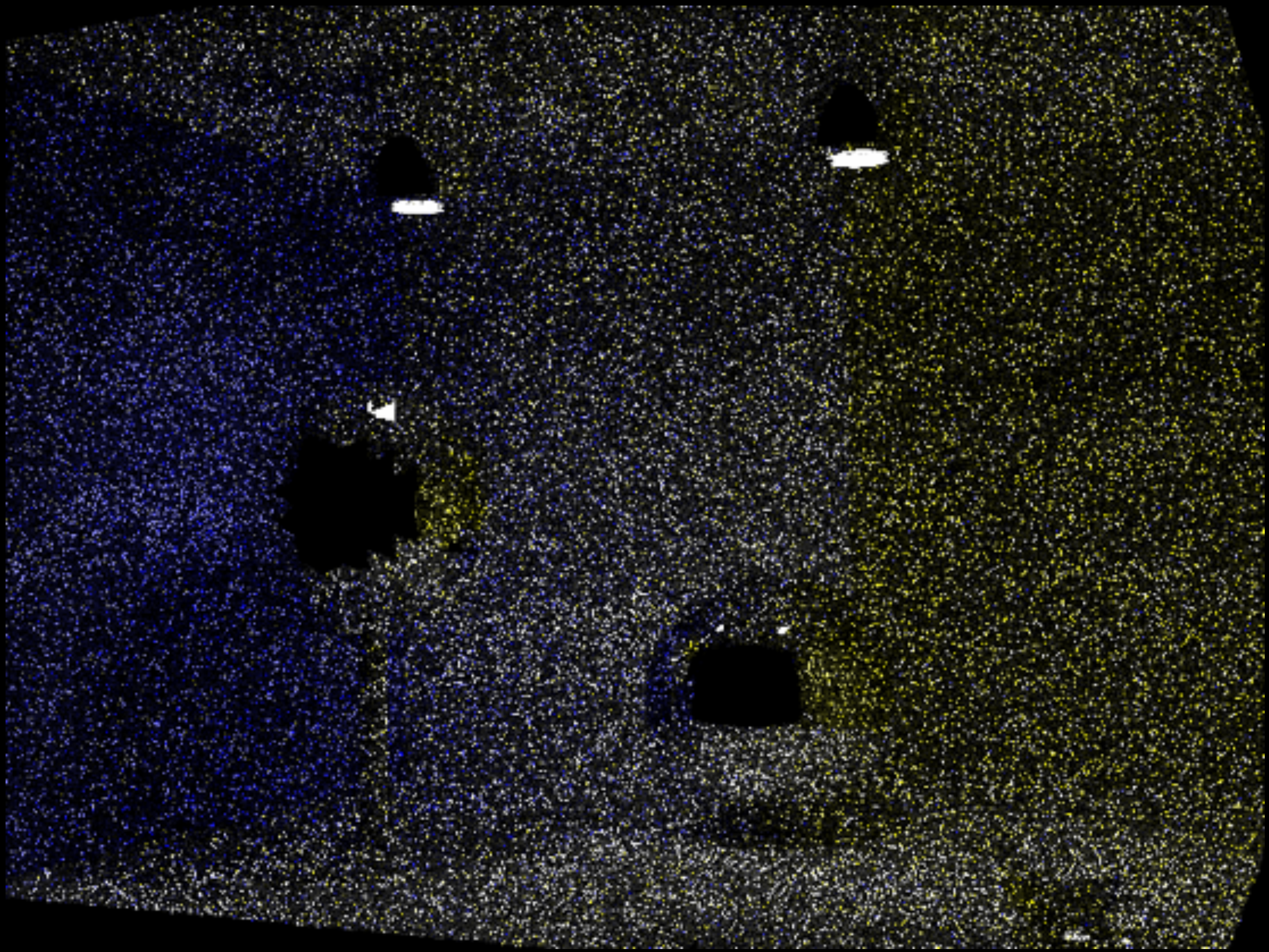
Accumulated flux $$\tau_{i+1} = \tau_i \frac{N_i + \alpha M_i}{N_i + M_i}$$

Radiance $$L_i = \frac{\tau_i}{\pi {R_i}^2} = \frac{\sum_{p=1}^{N_i} f_r \phi_p}{\pi {R_i}^2}$$

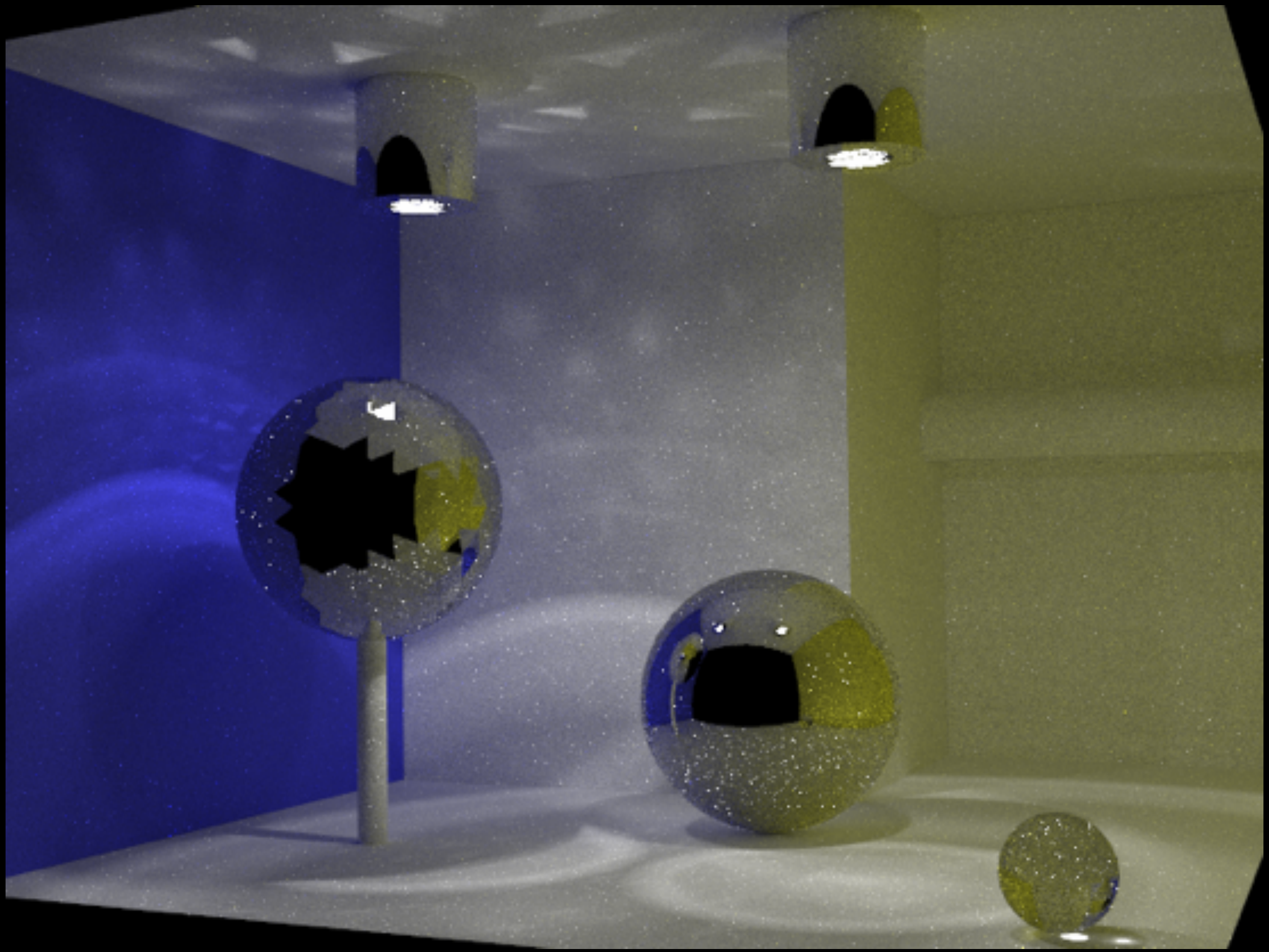
Equal-time Comparisons



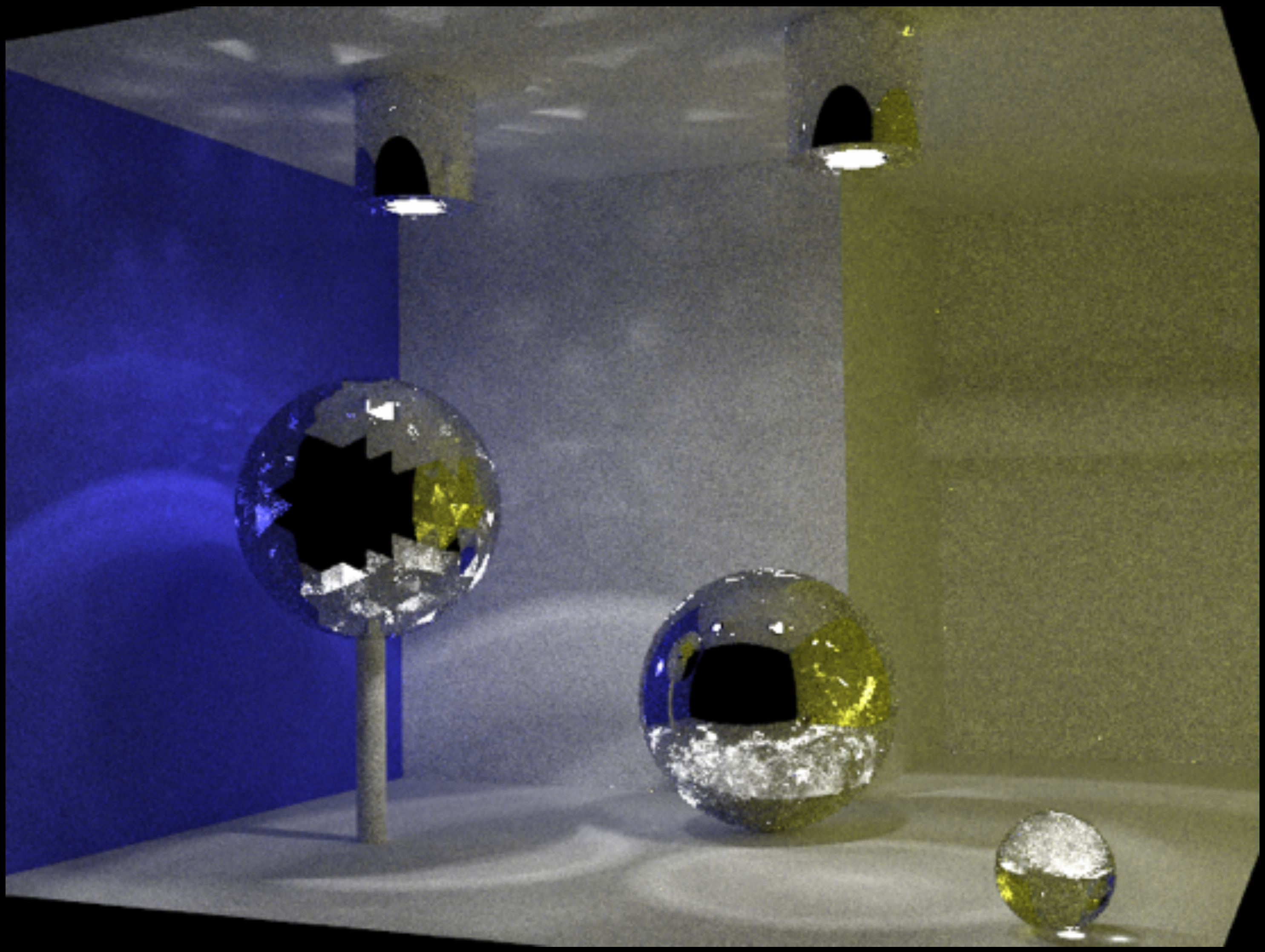
Path Tracing



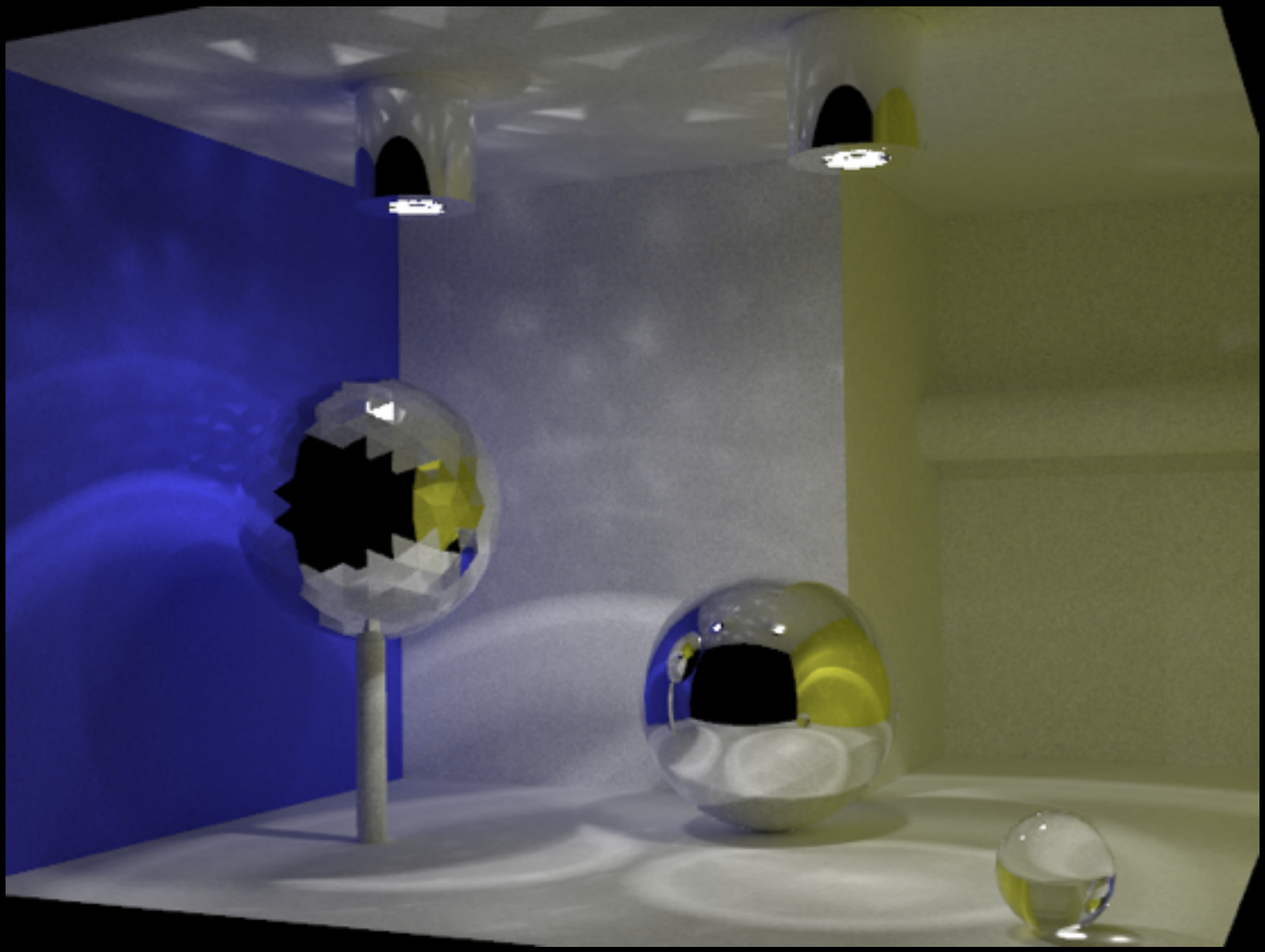
Bidirectional Path Tracing



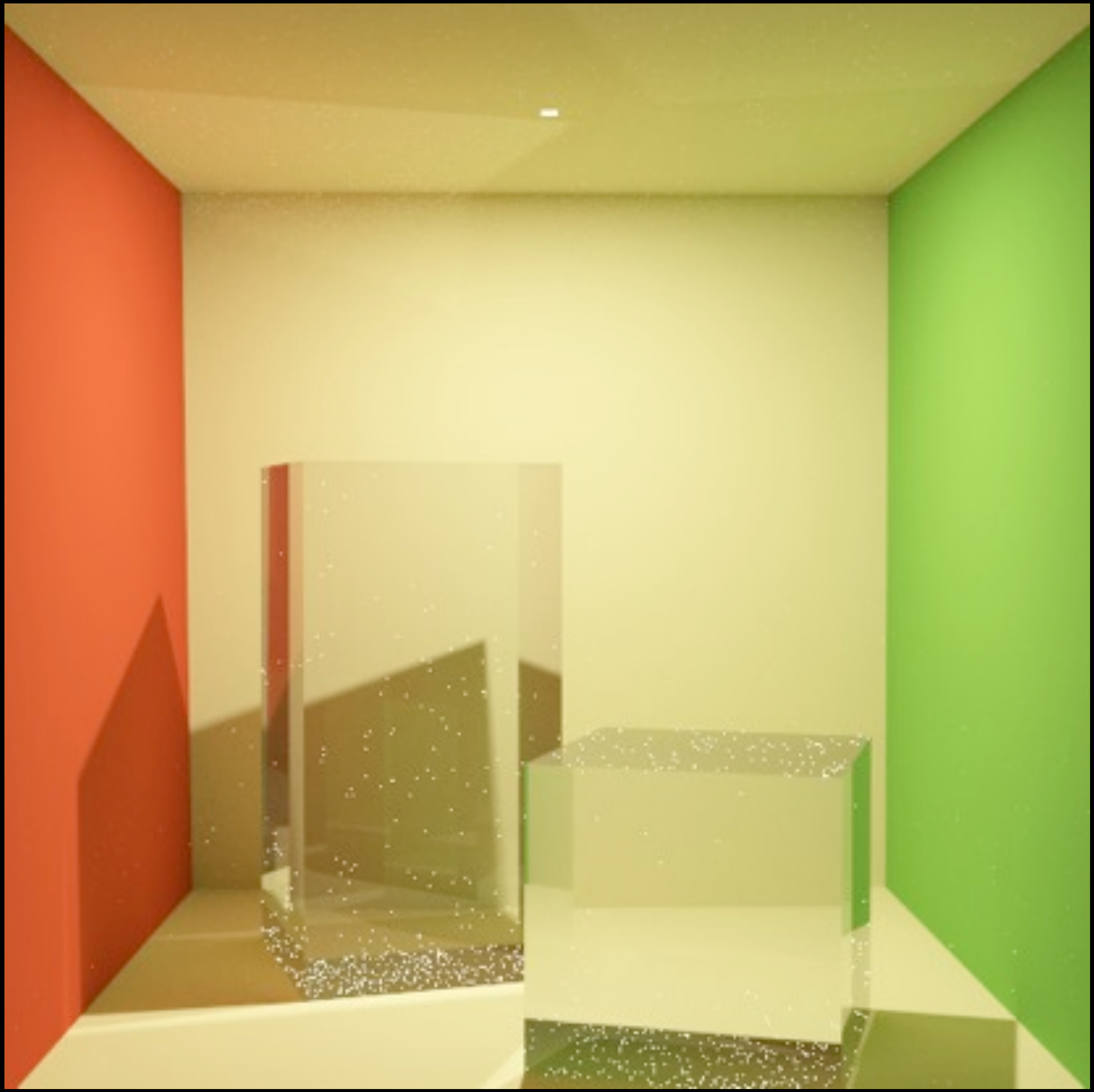
Metropolis Light Transport



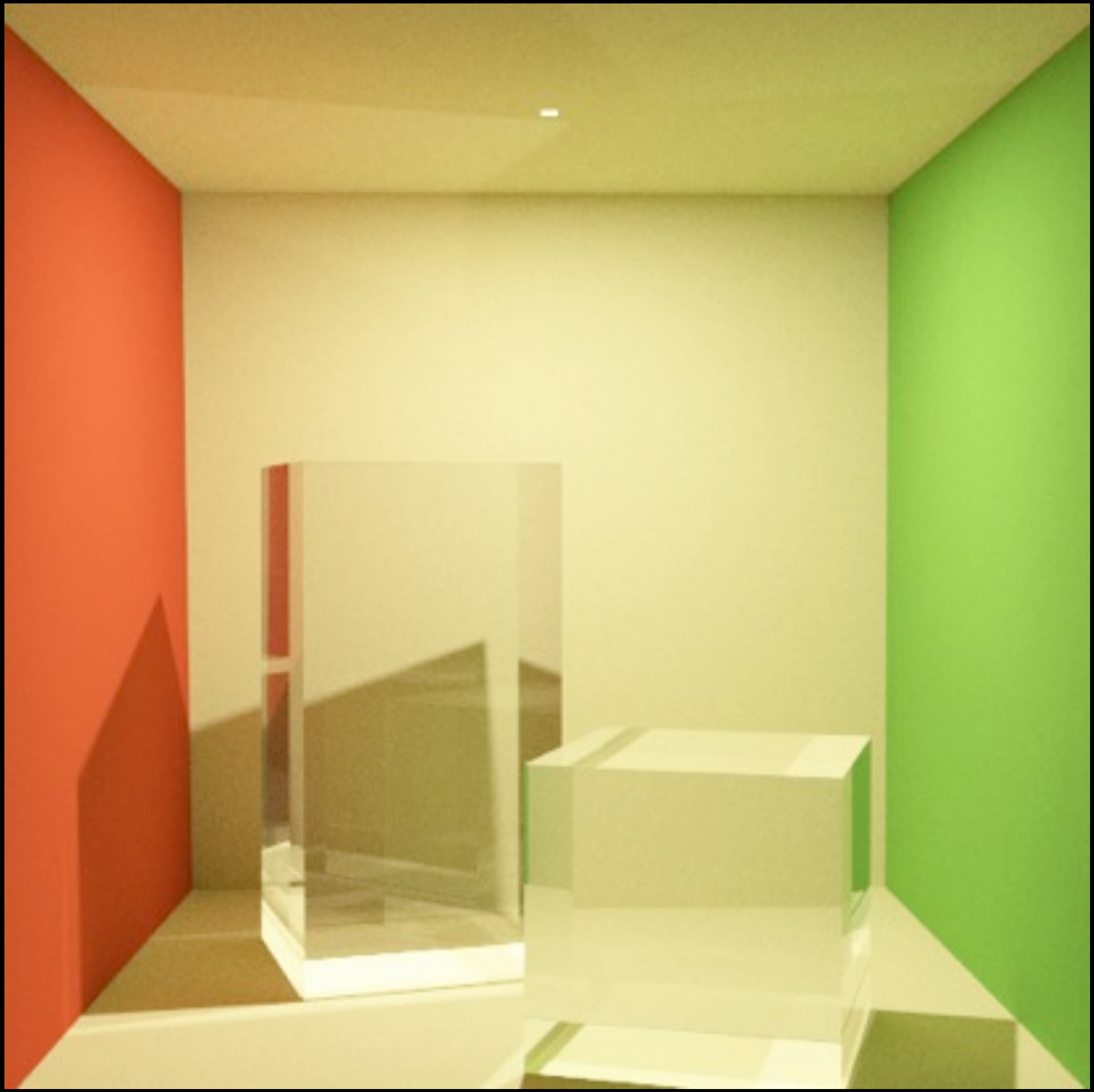
Progressive Photon Mapping



Bidirectional Path Tracing



Progressive Photon Mapping



Bidirectional Path Tracing

Progressive Photon Mapping

Metropolis Light Transport

Progressive Photon Mapping

Implementation

- Need an efficient range query of photons (*not* kNN query)
- Be aware of tradeoff
  - Tree is adaptive, but costly to construct
  - Hash is fast to construct, but not adaptive

Parameters

- Important parameters
  - Initial radius
  - Alpha value (= radius reduction rate)
  - Number of photons per pass
- Convergence is guaranteed no matter what (as long as $0 < \alpha < 1$)

Parameters - Initial Radius

- Several options
  - kNN queries on the first photon map
  - Ray differentials
  - Constant
- Avoid using too large radius
  - Few pixels per photons is usually good
- Anton will introduce a better approach

Parameters - Alpha Value

- Use 0.6... = 2/3
  - Theoretical optimal
  - Works well in practice
  - Adaptive alpha is possible

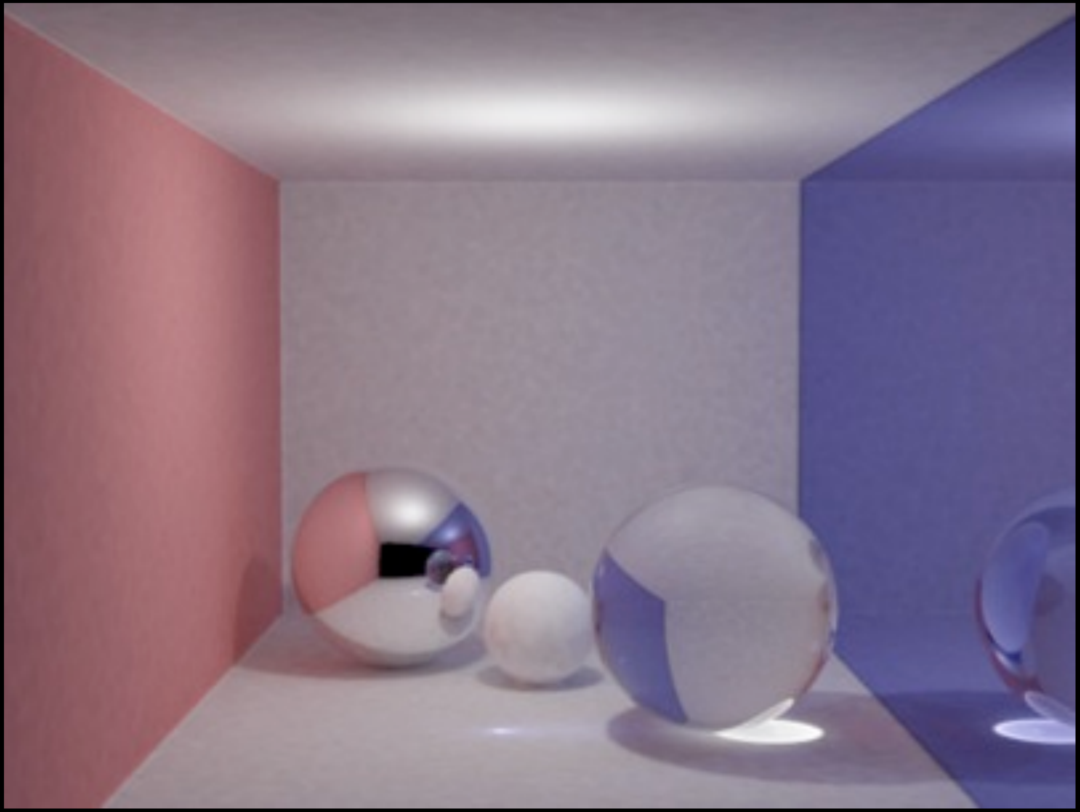
  - Anton will tell you more details

Parameters - # of Photons Per Pass

- Mainly depends on two factors
  - Desired frequency of visual feedback
  - Cost of making a data structure for range query
- Splatting approach can lower this number to one

Sample Code

- smallppm - 128 lines of working PPM code



cs.au.dk/~toshiya/smallppm.cpp

Sample Code

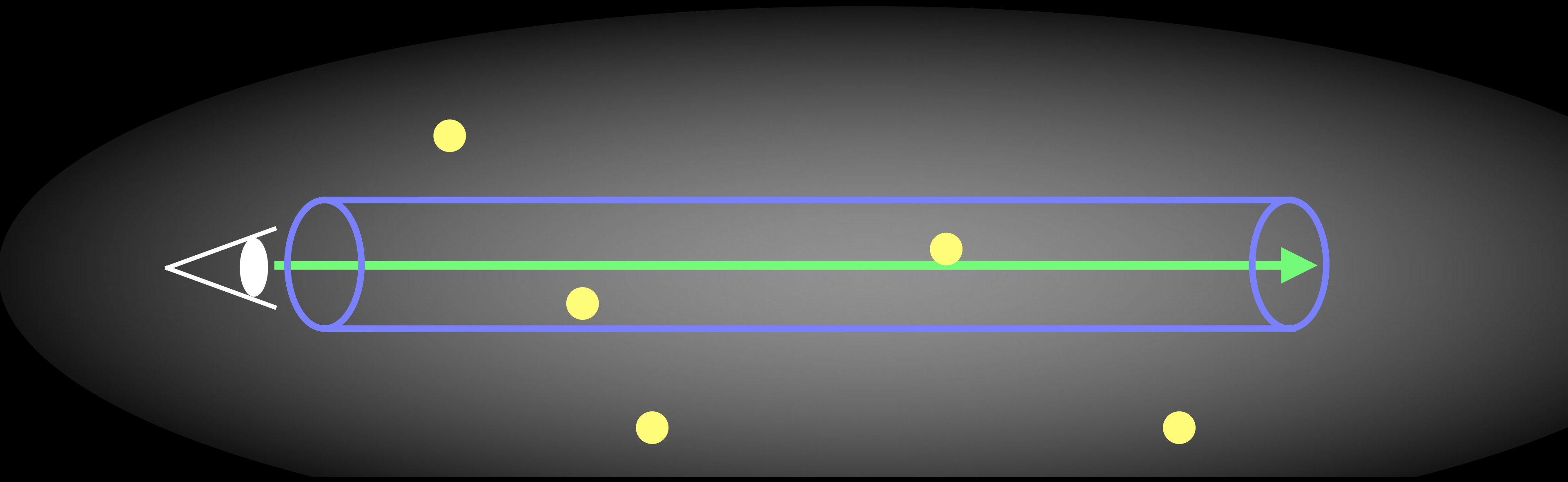
- Based on smallpt by Kevin Beason

- Splatting approach
- Uses spatial hashing for range queries

- Stores αN instead of N for each measurement point
  - Save some multiplications
  - αN is integer while N is not (avoids precision issue)

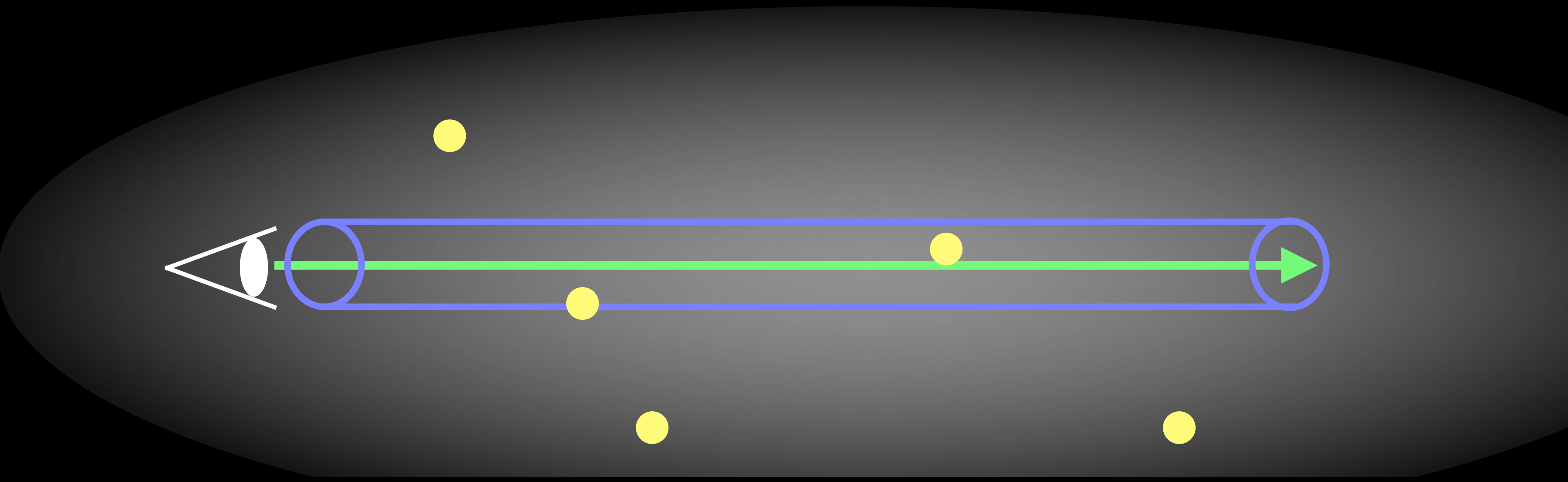
Participating Media

- Use a cylinder kernel around eye rays
  - Progressive extension of beam radiance estimate [Jarosz et al. 08]
  - Equations for radius reduction stay the same
  - More details in Section 5.6 of my dissertation



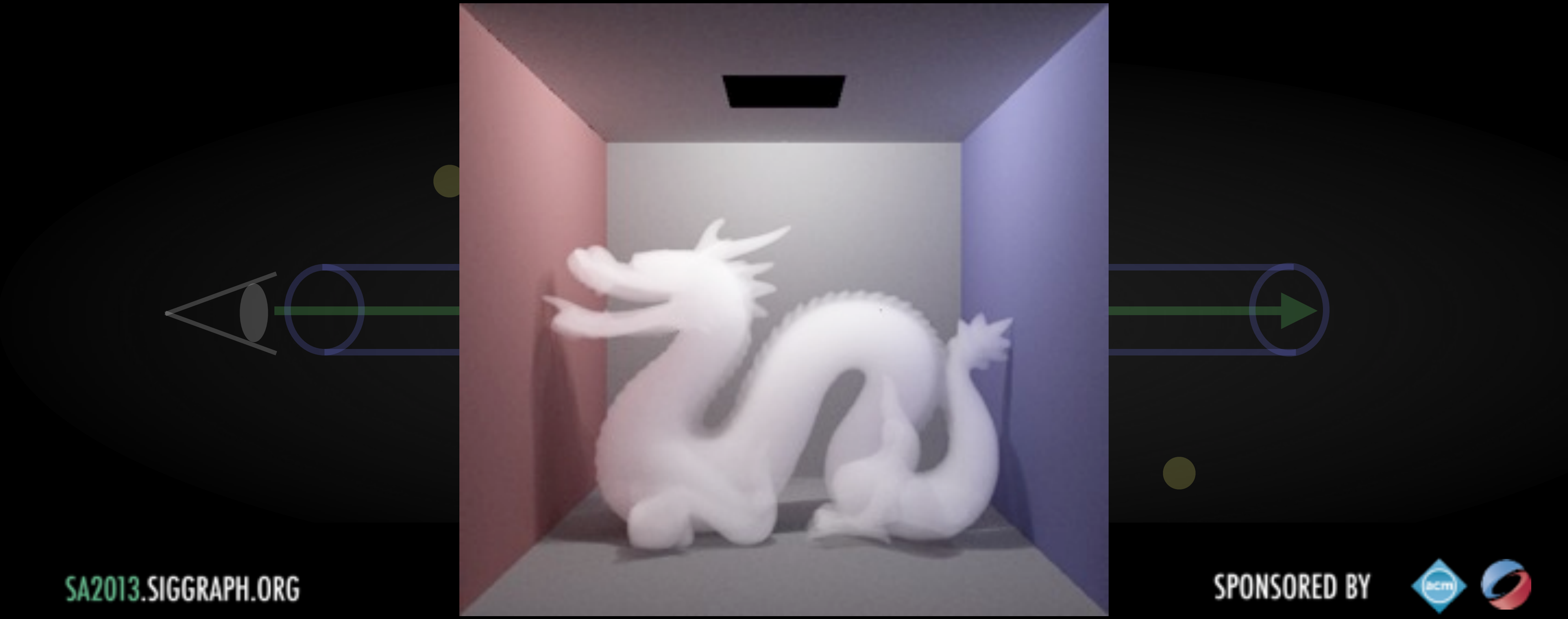
Participating Media

- Use a cylinder kernel around eye rays
  - Progressive extension of beam radiance estimate [Jarosz et al. 08]
  - Equations for radius reduction stay the same
  - More details in Section 5.6 of my dissertation



Participating Media

- Use a cylinder kernel around eye rays
  - Progressive extension of beam radiance estimate [Jarosz et al. 08]
  - Equations for radius reduction stay the same
  - More details in Section 5.6 of my dissertation



Summary

- Infinite number of photons without storing them
  - “Path tracing”nization of photon mapping

- Robust to specular-diffuse-specular paths
- Converges to the correct solution
- Easy to implement

PPM in the Wild

CHAOSGROUP v-ray

NVIDIA

VRED PROFESSIONAL

YAFARAY

LuxRender GPL PHYSICALLY BASED RENDERER

Next Talk

- Probabilistic formulation of PPM
- “How to turn your PM into PPM in a minute!”