

AI5505: Photorealistic Rendering

Global Illumination

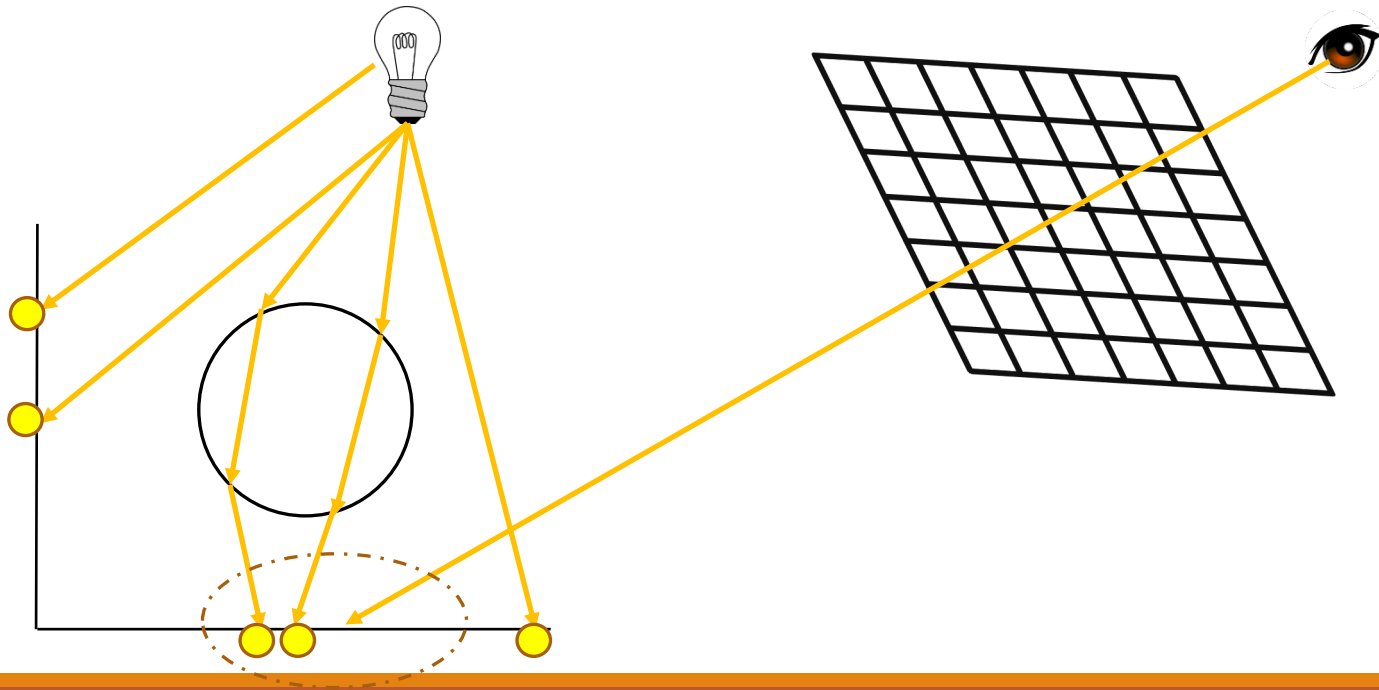
Lecturer: Bochang Moon

In the previous lecture,

- Photon Mapping (two pass rendering method)
 - 1. Generate photon maps
 - 2. Render an image using photon maps (radiance estimation)

Radiance Estimation

- $L_r(x, k_o) \approx \frac{1}{\pi r^2} \sum_{p=1}^N \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$

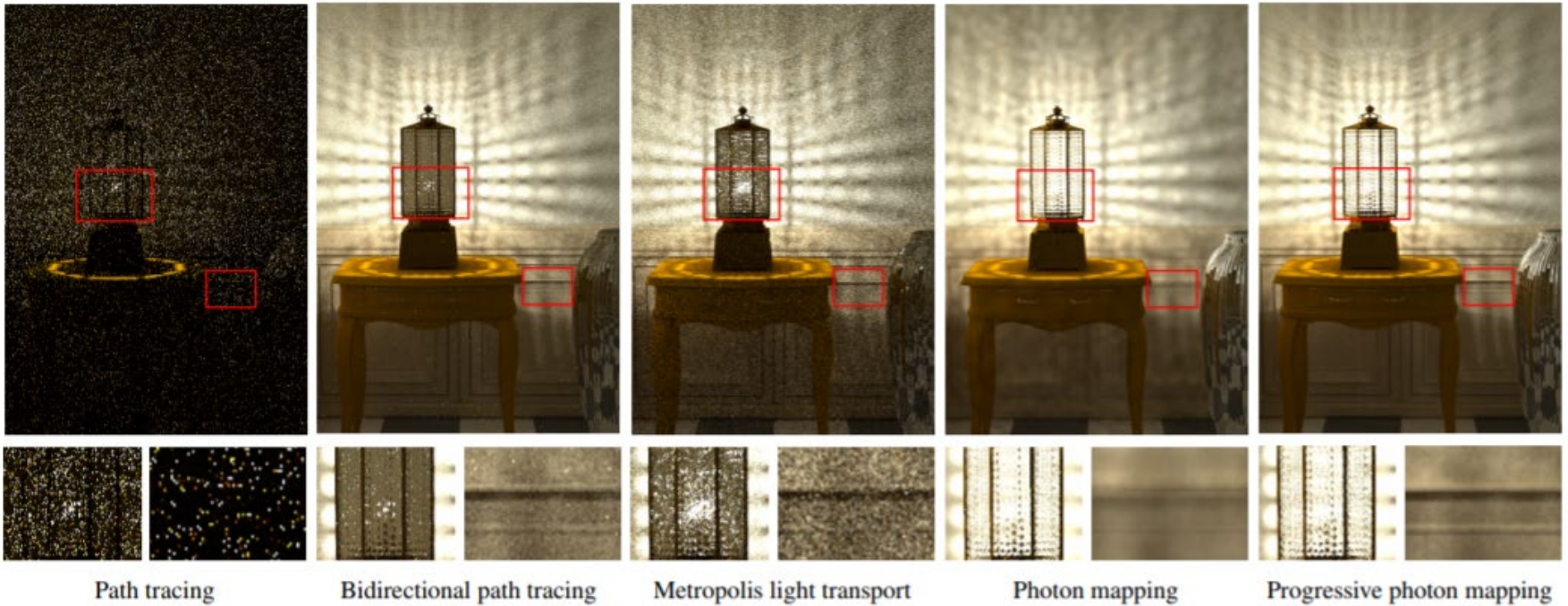


Radiance Estimation

- $L_r(x, k_o) \approx \frac{1}{\pi r^2} \sum_{p=1}^N \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$
- When $N \rightarrow \infty, r \rightarrow 0$, it converges to the correct solution
 - In practice, we cannot store the infinite number of photons due to a finite memory space

Progressive Photon Mapping

- SIGA 2008, Hachisuka et al.



Images from [Hachisuka et al. 08]

Progressive Photon Mapping

- Multi-pass rendering method
 - 1st pass
 - Query points are generated from the eye
 - Refinement passes:
 - Photon tracing
 - Progressive radiance estimation

Progressive Photon Mapping

- Photon Mapping

- $L_r(x, k_o) = \frac{1}{\pi r(x)^2} \sum_{p=1}^N \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$

- Progressive Photon Mapping

- $L_r^0(x, k_o) = \frac{1}{\pi r_0(x)^2} \sum_{p=1}^{N_0} \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$

- $L_r^1(x, k_o) = \frac{1}{\pi r_1(x)^2} \sum_{p=1}^{N_1} \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$

- ...

- $L_r^i(x, k_o) = \frac{1}{\pi r_i(x)^2} \sum_{p=1}^{N_i} \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$

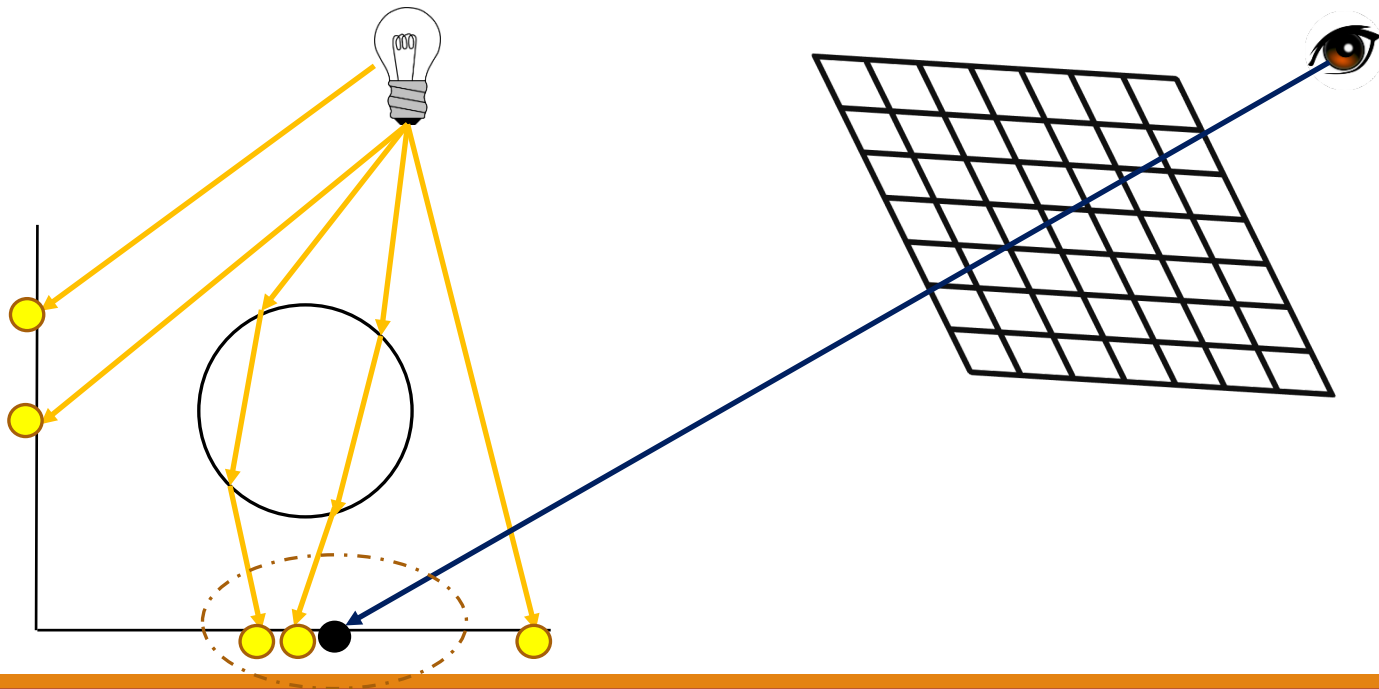
- ...

Progressive Photon Mapping

- Progressive Photon Mapping
 - $L_r^i(x, k_o) = \frac{1}{\pi r_i(x)^2} \sum_{p=1}^{N_i} \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$
 - $\lim_{i \rightarrow \infty} L_r^i(x, k_o) = L(x, k_o)$
 - Key properties:
 - $r_{i+1}(x) < r_i(x)$
 - $N_{i+1} > N_i$

Progressive Photon Mapping

- PPM assumes the photon density is locally uniform
- Reduce the radius of each hit point while accumulating newly added photons



Progressive Photon Mapping

- PPM assumes the photon density is locally uniform
- Reduce the radius of each hit point while accumulating newly added photons
- $\frac{N_i + M_i}{\pi r_i^2} = \frac{N_{i+1}}{\pi r_{i+1}^2}$
- $N_{i+1} = N_i + \alpha M_i$
 - N_i : # of photons in the previous steps
 - M_i : # of photons in the current step
 - α : a fraction of newly added photons to keep
- $\frac{N_i + M_i}{\pi r_i^2} = \frac{N_i + \alpha M_i}{\pi r_{i+1}^2}$
- $r_{i+1} = r_i \sqrt{\frac{N_i + \alpha M_i}{N_i + M_i}}$

Progressive Photon Mapping

- Flux correction

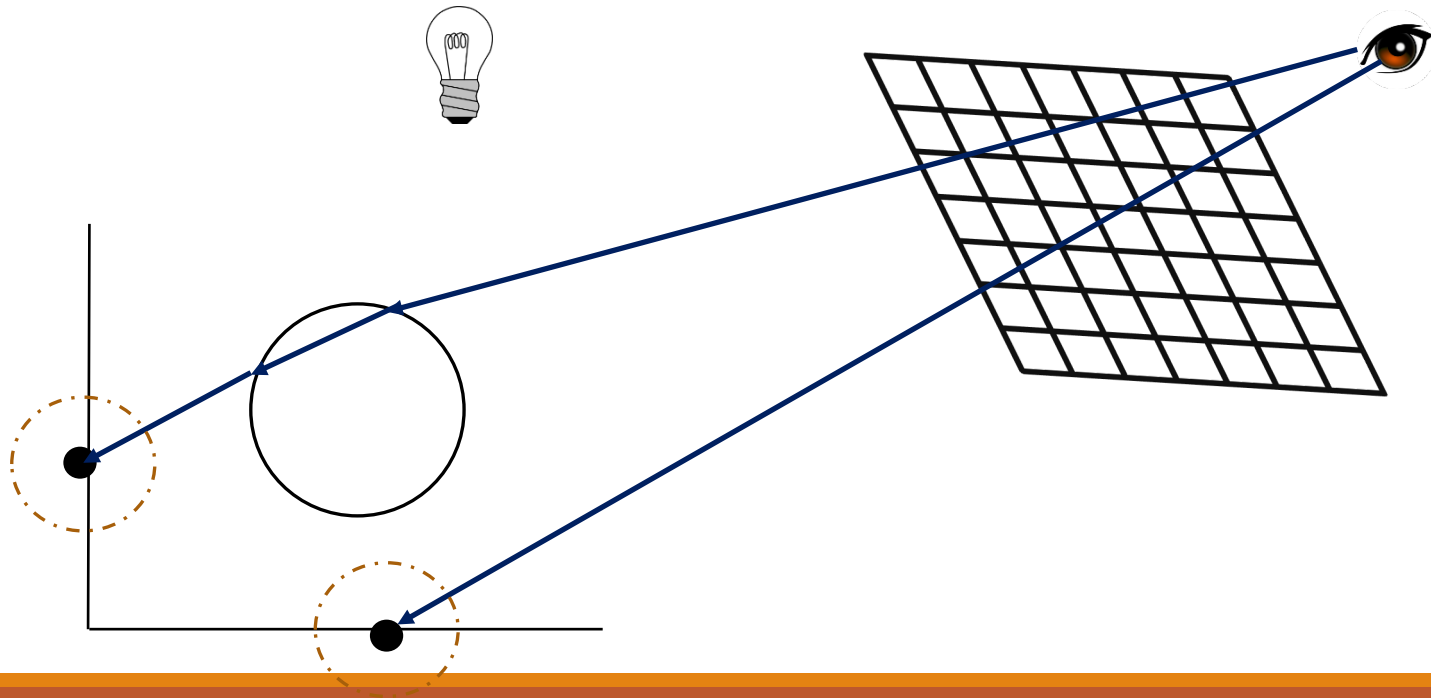
- $\tau_{N_i}(x, k_o) = \sum_{p=1}^{N_i} \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$
- $\tau_{M_i}(x, k_o) = \sum_{p=1}^{M_i} \rho(x, k_p, k_o) \Delta\Phi_p(x, k_p)$
- $\tau_{N_{i+1}}(x, k_o) = \left(\tau_{N_i}(x, k_o) + \tau_{M_i}(x, k_o) \right) \frac{N_i + \alpha M_i}{N_i + M_i}$

- Radiance

- $L_r^i(x, k_o) = \frac{1}{\pi r_i^2} \times \frac{\tau_{N_i}}{\text{total \# of emitted photons}}$

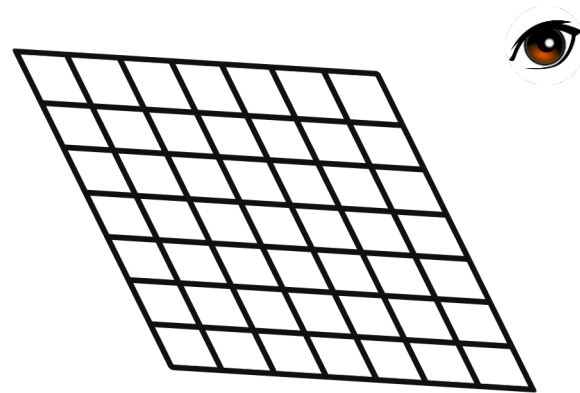
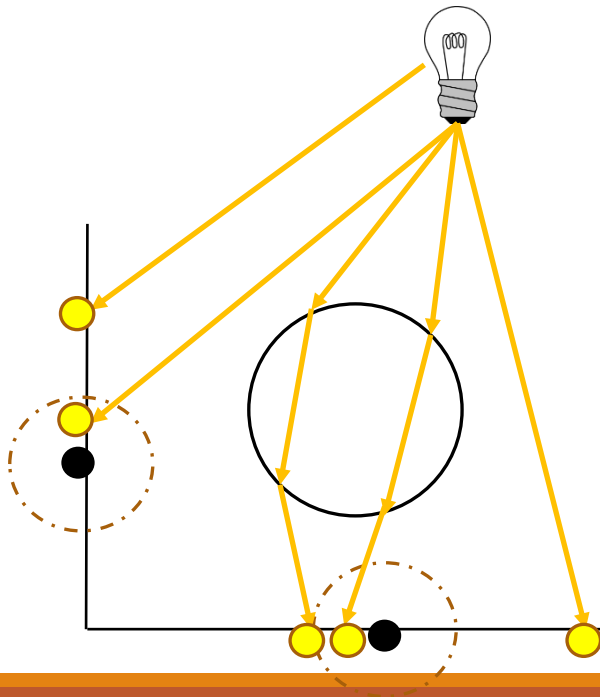
Progressive Photon Mapping

- 1st pass
 - Generate hit points



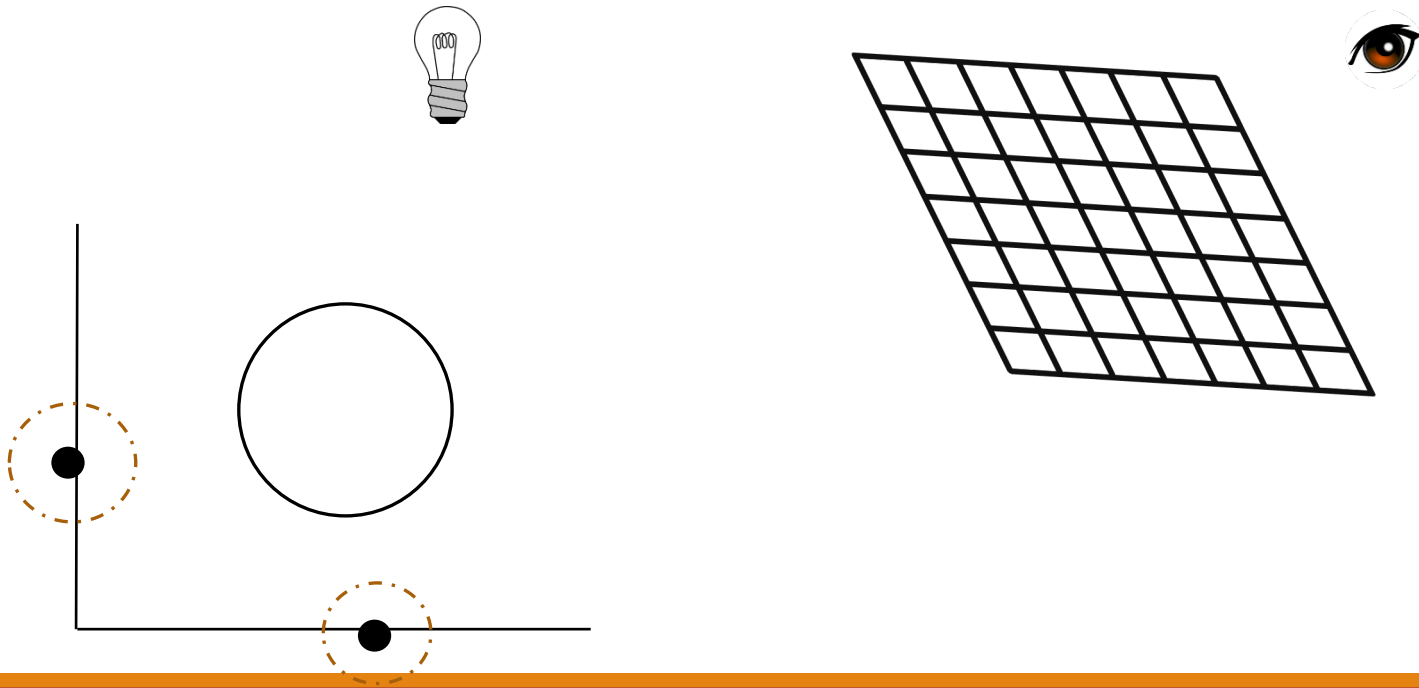
Progressive Photon Mapping

- 1st Refinement pass
 - Generate photons



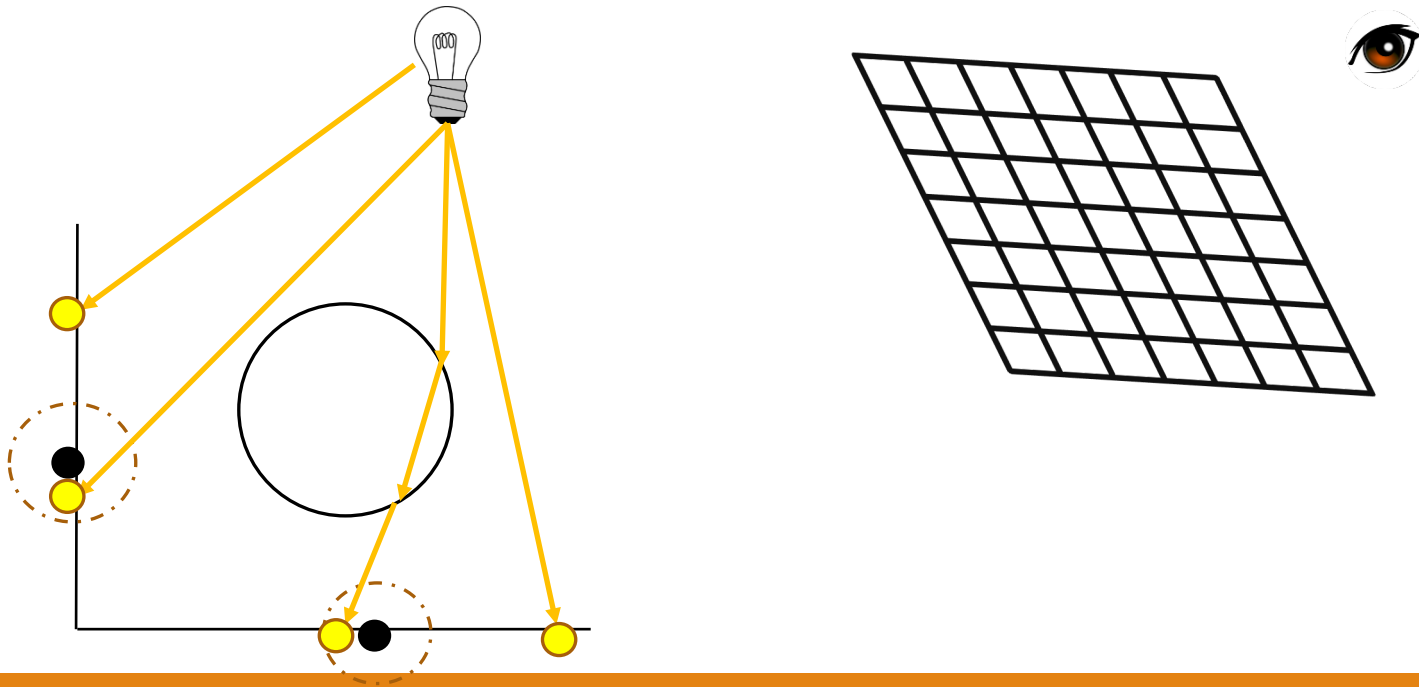
Progressive Photon Mapping

- 1st Refinement pass
 - Generate photons



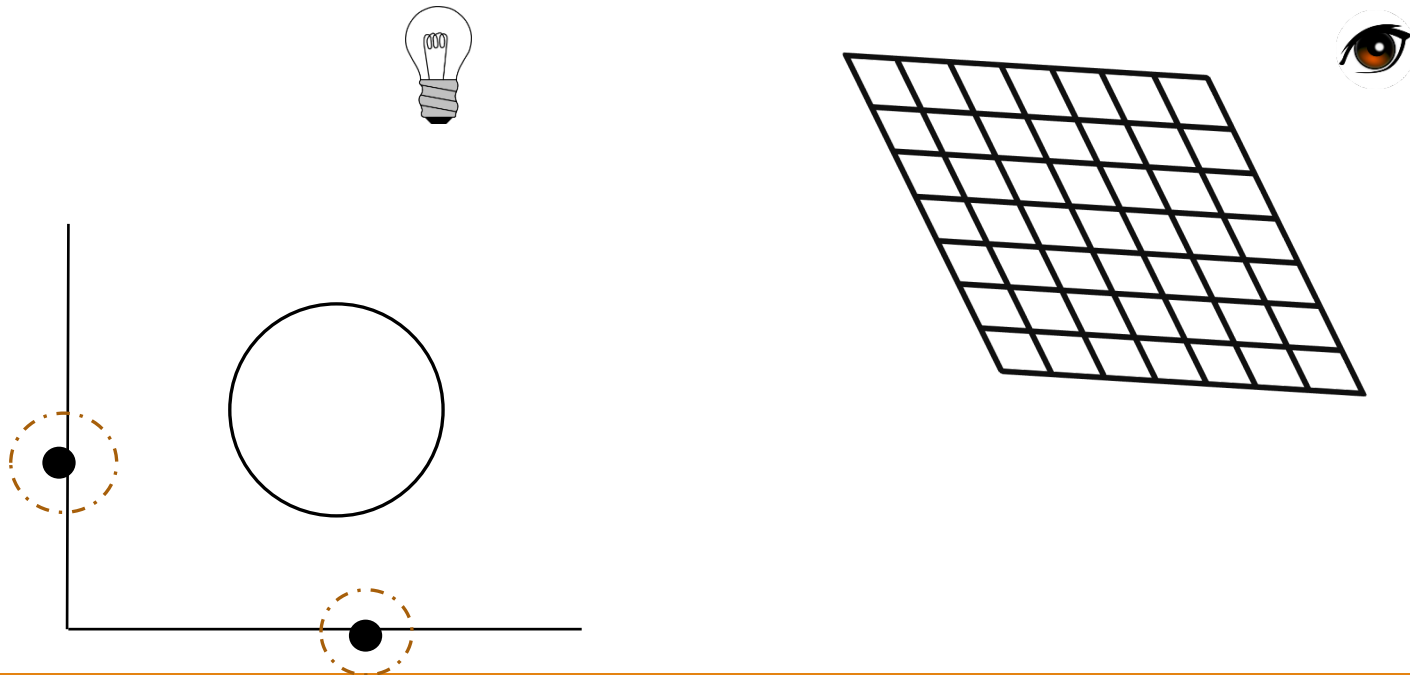
Progressive Photon Mapping

- 2nd Refinement pass
 - Generate photons



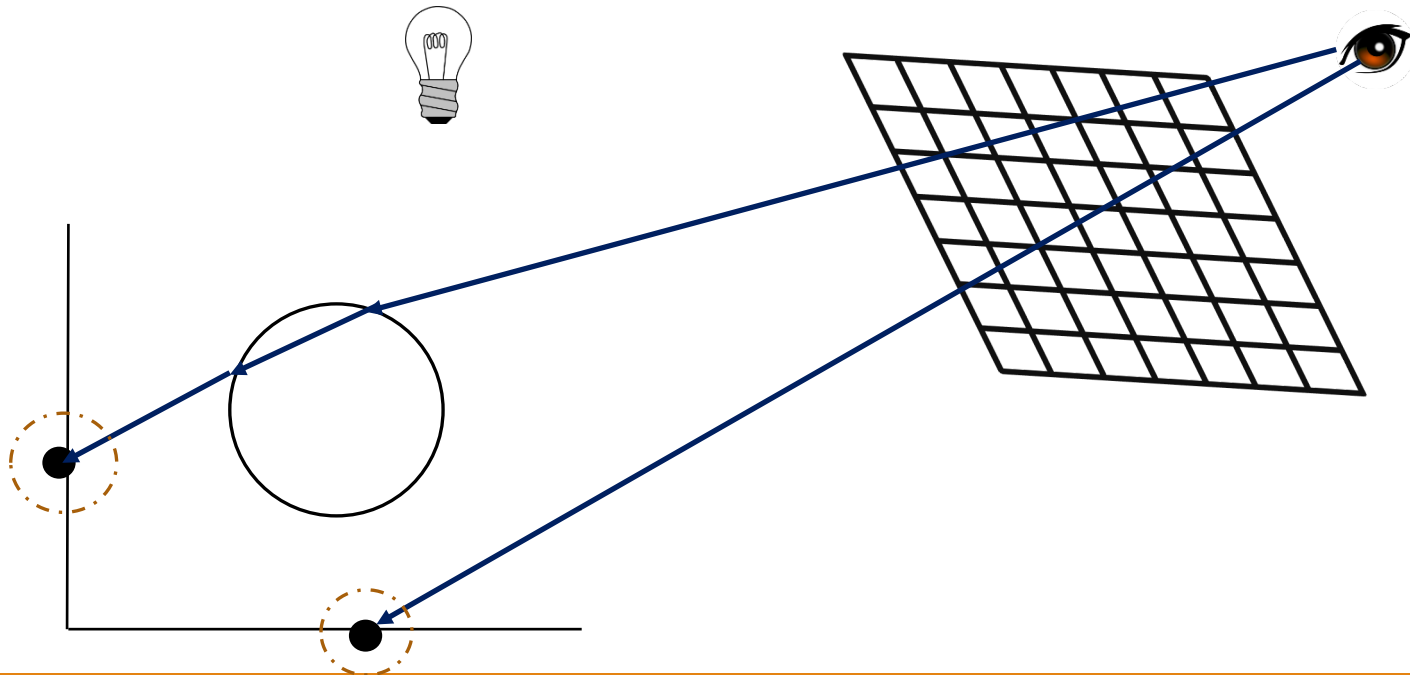
Progressive Photon Mapping

- 2nd Refinement pass
 - Generate photons

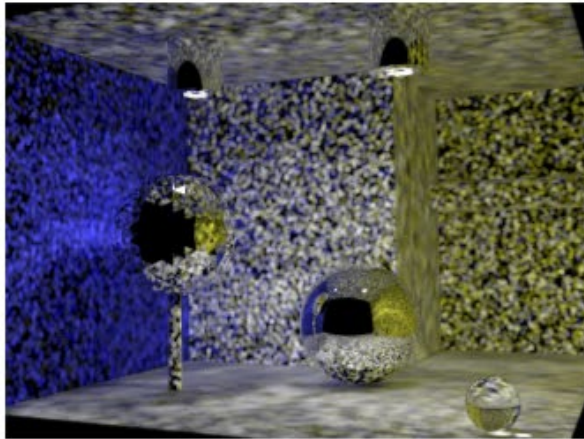


Progressive Photon Mapping

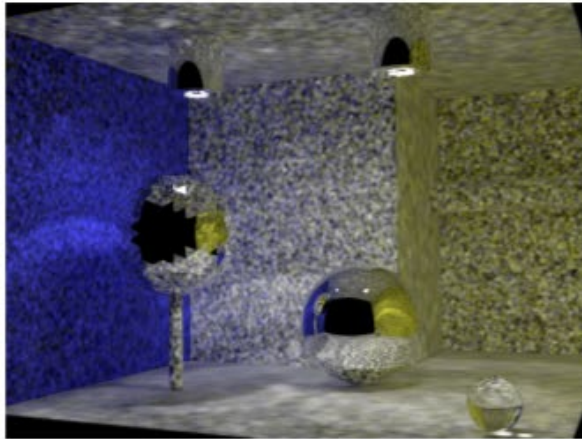
- Rendering



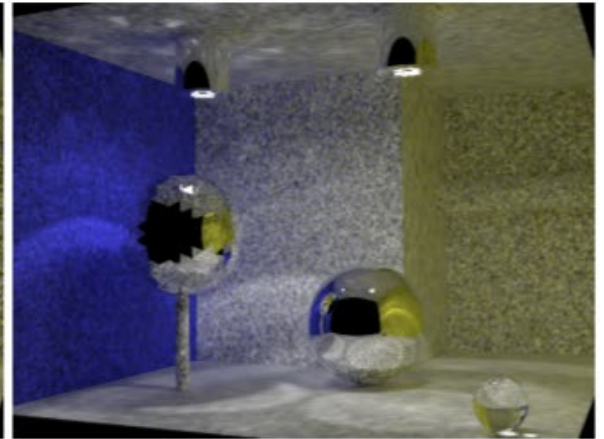
PPM Results



0.1M photons

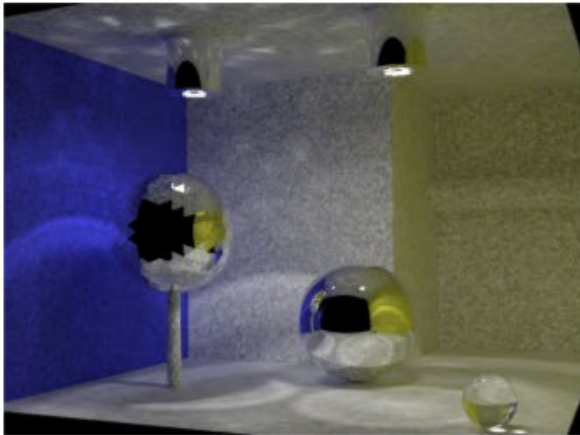


0.4M photons

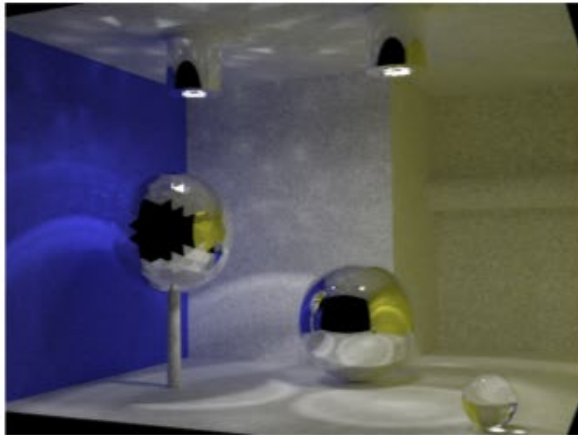


1.6M photons

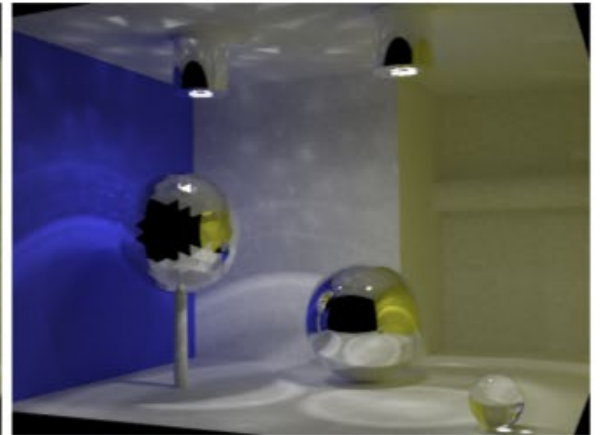
PPM Results



6.4M photons

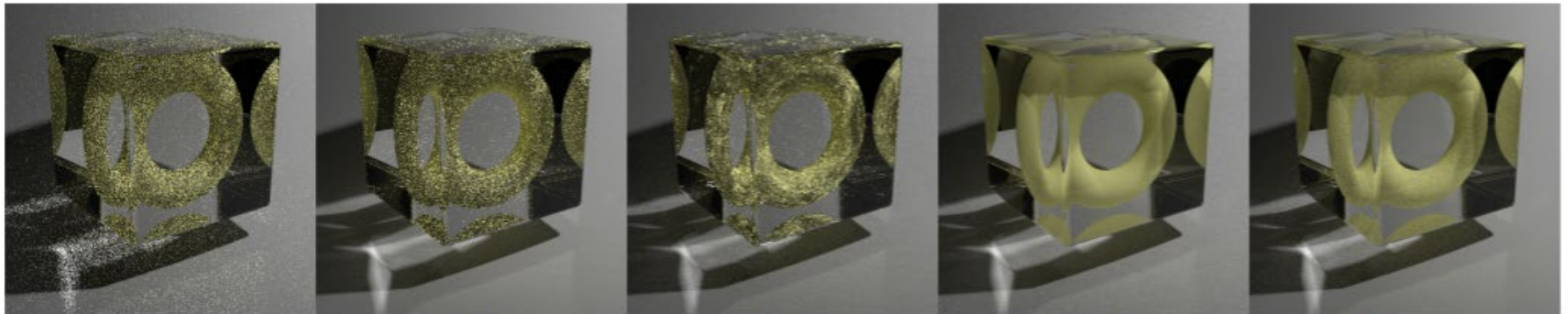


25.6M photons



102.4M photons

PPM Results



PT

BDPT

MLT

PPM

Reference

Further Reading

- Distributed effects?
 - # of hit points can introduce a memory issue especially for distributed effects
 - Stochastic Progressive Photon Mapping, Hachisuka et al., SIGA09
- Progressive error estimation
 - A Progressive Error Estimation Framework for Photon Density Estimation, Hachisuka et al., SIGA10
- An alternative to PPM
 - Progressive Photon Mapping: A Probabilistic Approach, Knaus and Zwicker, TOG11