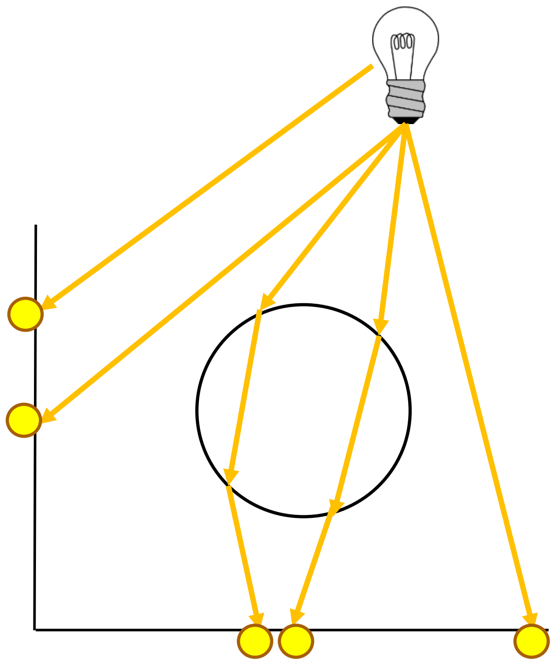
CT5503: Photorealistic Rendering

Global Illumination

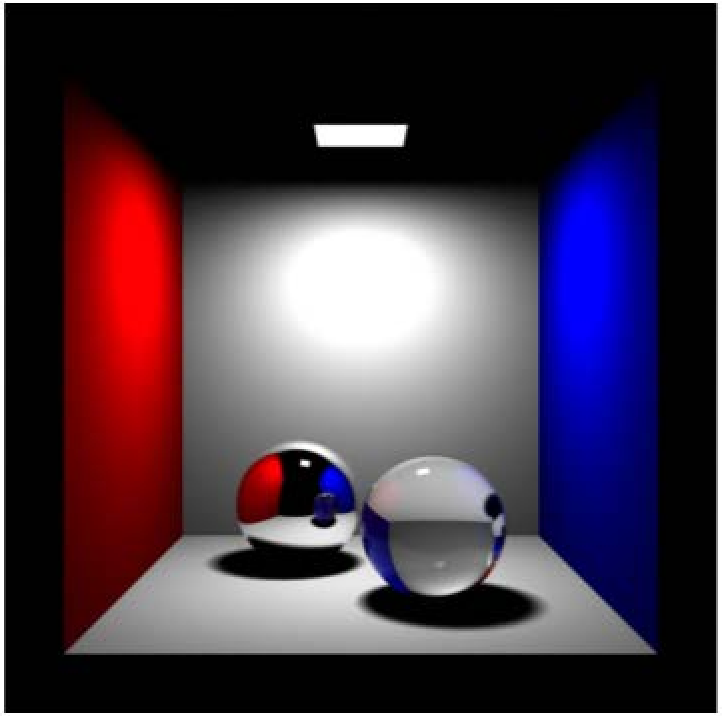
BOCHANG MOON

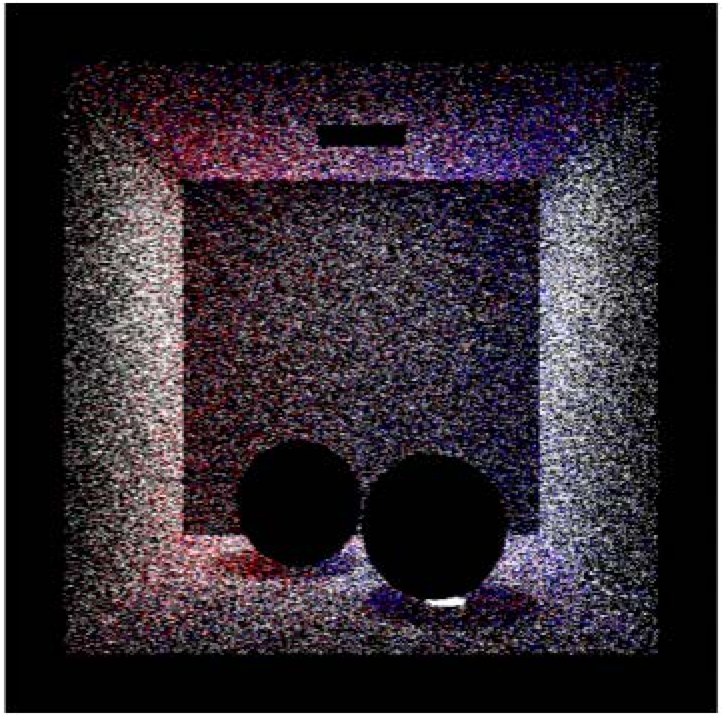
Photon Mapping

- [Jensen 1996]
- A two-pass rendering method
  - 1. build a photon map
  - 2. render an image using the map



Photon Mapping

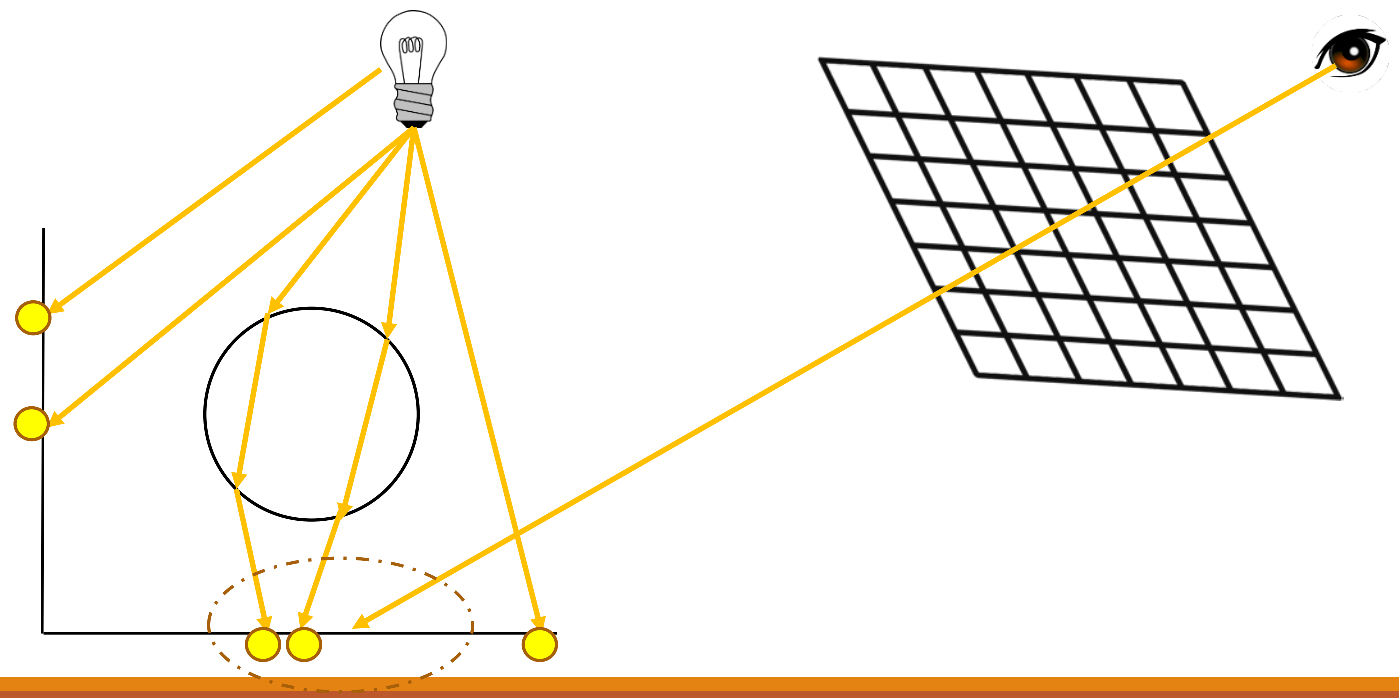




[Christensen and Jensen, SIGGRAPH 2000 course]

Photon Mapping

- [Jensen 1996]
- A two-pass rendering method
  - 1. (photon tracing) build a photon map
  - 2. (radiance estimation) render an image using the map



Photon Mapping

- Photon emission
  - Photons are generated from light sources
  - Each photon carries a fraction of the power of a light source
- Photon
  - Position (x, y, z)
  - Power $\Phi_{photon} = \frac{\Phi_{light}}{n_e}$
  - Incident direction

Photon Emission

- Photon emission
  - $n_e = 0$
  - $for\ (i = 0\ to\ N)$
    - $select\ a\ random\ direction\ d$
    - $trace\ a\ photon\ from\ a\ light\ position\ p\ in\ direction\ d$
    - $n_e = n_e + 1$
  - $scale\ power\ of\ stored\ photons\ with\ \frac{1}{n_e}$
- Multiple light sources?
  - Each light source emits photons
  - Brighter light sources can emit more photons than from dim ones

Photon Scattering

- Photon tracing employs a ray tracing procedure:
  - If a photon hits a surface, it will be reflected or absorbed

- In practice, a Russian Roulette (RR) is used
  - RR is a stochastic technique to determine whether a photon is reflected or not
  - RR is widely used in Monte Carlo ray tracing methods
  - It improves efficiency by increasing likelihood that samples can have high contributions

Photon Scattering

- Photon tracing employs a ray tracing procedure:
  - If a photon hits,
    - $p = d$ $\quad$ $(probability\ of\ reflection = reflectivity\ of\ the\ surface)$
    - $\xi \in [0,1]$ $\quad$ $(uniformly\ distributed\ random\ number)$
    - $if\ (\xi < p\ )$
      - $reflect\ photon\ with\ power\ \Phi_p$
    - $else$
      - $photon\ is\ absorbed$

- Note
  - If N photons hit, we have two choices
    - Reflect N photons with half the power
    - Reflect N/2 photons with the full power (RR enables this efficient approach)

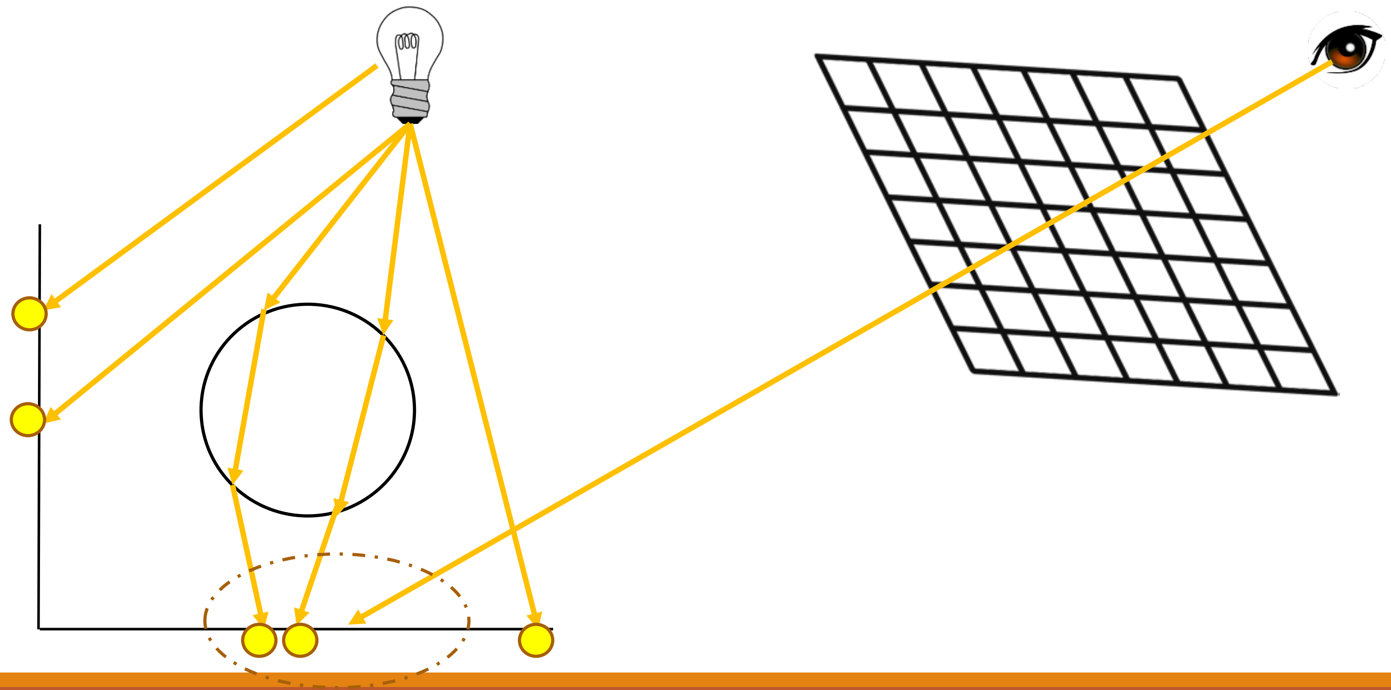
Photon Storing

- Store photons when they hit non-specular surfaces
  - Do not need to store them for specular reflection (e.g., reflection on mirrors)

- An emitted photon can be stored several times along its path
- A tree structure (kd-trees) is used to maintain the photons
  - This will be utilized for searching neighboring photons in the second step

Photon Mapping

- [Jensen 1996]
- A two-pass rendering method
  - 1. (photon tracing) build a photon map
  - 2. (radiance estimation) render an image using the map



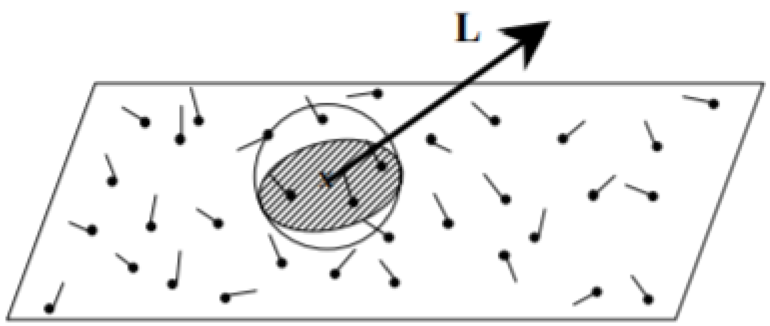
Radiance Estimation

- Need to estimate photon density $\frac{d\Phi}{dA}$ at query points (e.g., intersection points between primary rays and surfaces)
- $L_r(x, k_o) = \int_{\Omega_x} \rho(x, k_i, k_o) L_i(x, k_i)(n_x \cdot k_i) dk_i$
  - $L_i(x, k_i) = \frac{d^2\Phi_i(x,k_i)}{(n_x \cdot k_i) dk_i dA_i}$
- $L_r(x, k_o) = \int_{\Omega_x} \rho(x, k_i, k_o) \frac{d^2\Phi_i(x,k_i)}{(n_x \cdot k_i) dk_i dA_i} (n_x \cdot k_i) dk_i$
- $= \int_{\Omega_x} \rho(x, k_i, k_o) \frac{d^2\Phi_i(x,k_i)}{dk_i dA_i}$
- The incoming flux, $\Phi_i(x, k_i)$, is approximated with the photon map
  - Search k nearest photons from x

Radiance Estimation

- $L_r(x, k_o) = \int_{\Omega_x} \rho(x, k_i, k_o) \frac{d^2\Phi_i(x,k_i)}{dk_i dA_i}$
- Each photon *p* carries a fraction of a light power, $\Delta\Phi_p(k_p)$
- $L_r(x, k_o) \approx \sum_{p=1}^{N} \rho\left(x, k_p, k_o\right) \frac{\Delta\Phi_p(x,k_p)}{\Delta A}$
- $\Delta A = \pi r^2$
  - The radius r is set as the largest distance between x and positions of photons
  - Assume a locally flat surface around x

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



[Christensen and Jensen, SIGGRAPH 2000 course]

Radiance Estimation

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

- When assumptions (e.g., locally flat surfaces) are valid and the number of photons is infinite, the approximation error will be zero

- Q. can we store the infinite number of photons?
  - If not, is there any way to accomplish the consistency?
  - As an advanced topic in this direction, we will study Progressive Photon Mapping which addresses this issue