Progressive Photon Mapping

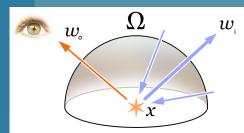
T. Hachisuka, S. Ogaki and H. Jensen

Presented by: Garoe Dorta Perez

CM50245: Computer Animation and Games II

- Rendering equation
- $L_o(x, \omega_o) = L_e(x, \omega_o) +$

$$L_e(x,\omega_o) + \int_{\Omega} f(x,\omega_o,\omega_i) L_i(x,\omega_i) |cos heta_i| d\omega_i$$

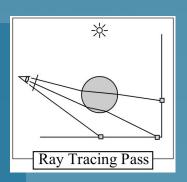


• Photon mapping as an approximation

- Two passes
 - 1. Ray tracing in a photon map
 - 2. Photon rendering
- $\overline{|\bullet|} L_r(x,\omega_o) pprox \sum_{p=1}^N rac{f(x,\omega_o,\omega_i)\phi_p(x_p,\omega_i)}{\pi r^2}$

Ray tracing pass 1

- Building photon map
- Ray tracing to find visible surfaces
- Each ray includes all specular bounces
- Stop when non-specular surface is found



• Ray tracing pass 2

Struct hitPoint

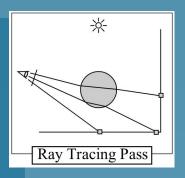
x hit position

i,j pixel location

R current radius

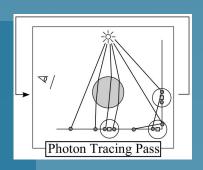
N acum photon count

au acum reflected flux



Photon tracing pass

- Accumulate photon flux in hit points
- New photons improve the quality
- $d(x) = \frac{n}{\pi r^2}, \ d'(x) = \frac{n'}{\pi r^2}$



Radius reduction

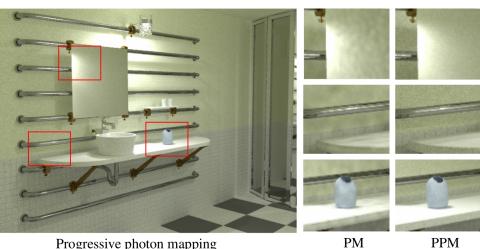
- Radius reduces with each pass to increase quality
- There has to be a gain in total number of photons
- Keep fraction of new photons $\hat{N}(x) = N(x) + \alpha M(x)$
- New radius $\hat{R}(x) = R(x) \left(\frac{N(x) + \alpha M(x)}{N(x) + M(x)} \right)^{\frac{1}{2}}$

Flux correction

- Reduced radius = incorrect accumulated flux
- $\tau_N(x,\omega_o) = \sum_{p=1}^{N(x)} f(x,\omega_o,\omega_i) \phi_p'(x_p,\omega_i)$
- $\bullet \tau_{\hat{N}}(x,\omega_o) = (\tau_N(x,\omega_o) + \tau_M(x,\omega_o)) \frac{N(x) + \alpha M(x)}{N(x) + M(x)}$

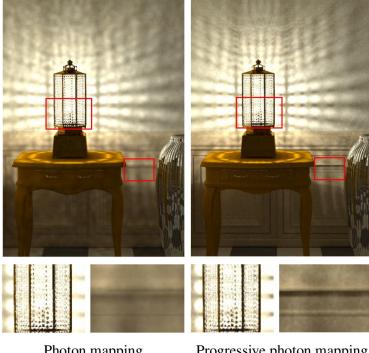
Radiance evaluation

- Sum the contribution of all photons in the hit point
- $L(x, \omega_o) = \frac{\tau(x, \omega_o)}{\pi R(x)^2 N_{em}}$



Progressive photon mapping

PPM



Photon mapping Progressive photon mapping

Possible improvements

- Locally adaptive
- Optimal number of total photon
- Optimal direction direction
- Optimal intensity

