State of the Art in Photon Density Estimation
Photon Mapping Basics

Wojciech Jarosz
Motivation - Global Illumination
Motivation - Direct Illumination
Motivation - Before photon mapping

- Radiosity
  - ❌ Mostly diffuse
  - ❌ Mesh based lighting representation

- Monte Carlo path tracing
  - ✔ Very general
  - ❌ Noisy
  - ❌ Computation time/slow convergence
Path Tracing

10 paths/pixel

Henrik Wann Jensen
Path Tracing Caustics

10 paths/pixel

Henrik Wann Jensen
Path Tracing Caustics
Path Tracing Caustics
Path Tracing Caustics
Density Estimation

- Light tracing / “Backward ray tracing”
- James Arvo. In *Developments in Ray Tracing*, SIGGRAPH ‘86 Course Notes
Light Tracing ("Backward" Ray Tracing)

Preprocess:
- parametrize surfaces and create "illumination maps"
- shoot light from light sources
- deposit photon energy in illumination maps
For each shading point

- compute direct lighting
- lookup indirect lighting + caustics from illumination maps
Light Tracing

✅ One of first techniques to simulate caustics

❌ Requires parametrizing surface or meshing
  - Complex/procedural geometry difficult to handle

❌ Illumination map resolution difficult to choose
  - high resolution/few photons: high-frequency noise
  - low resolution/many photons: blurred illumination
Photon Mapping

A two-pass algorithm:

- Pass 1: Trace virtual photons from the light source, scatter at surfaces, and cache
- Pass 2: Ray trace the scene and use the photons to compute indirect illumination

Same as light tracing, but different way of storing/caching photons
Photon Mapping (Photon Tracing)

1) Emit photons
2) Scatter photons
3) Store photons
Photon Mapping (Photon Tracing)

1) Emit photons
2) Scatter photons
3) Store photons
Photon Mapping (Photon Tracing)

1) Emit photons
2) Scatter photons
3) Store photons
Photon emission

Define initial:

- $x_p$ : position
- $\omega_p$ : direction
- $\Phi_p$ : photon power

Recipe:

- Sample position on surface area of light $x_p \sim \text{pdf}(x_p)$
- Sample direction $\omega_p \sim \text{pdf}(\omega_p)$
- $\Phi_p = L_e(x_p, \omega_p) / \text{pdf}(x_p) / \text{pdf}(\omega_p)$
Photon emission examples

- **Isotropic point light:**
  - Generate uniform random direction over sphere

- **Spotlight:**
  - Generate uniform random direction within spherical cap

- **Diffuse square light**
  - Generate uniform random location on square
  - Generate cosine-weighted direction over hemisphere
void generatePhotonMap()
    repeat:
        \((l, \text{Prob}_l) = \text{chooseRandomLight}()\)
        \((x, \omega, \Phi) = \text{emitPhotonFromLight}(l)\)
        \[\text{tracePhoton}(x, \omega, \Phi / \text{Prob}_l)\]
    until we have enough photons;
    divide all photon powers by number of emitted photons
void generatePhotonMap()
    repeat:
        (l, Prob_l) = chooseRandomLight()
        (x, ω, Φ) = emitPhotonFromLight(l)
        tracePhoton(x, ω, Φ / Prob_l)
    until we have enough photons;
    divide all photon powers by number of emitted photons

void tracePhoton(x, ω, Φ)
void tracePhoton(x, \omega, \Phi)
    s = nearestSurfaceHit(x, \omega)
    x += s*\omega    // propagate photon
    possiblyStorePhoton(x, \omega, \Phi)
    (\omega', \text{pdf}) = sampleBxDF(x, \omega)
    return tracePhoton(o, \omega', \Phi * BxDF(x, \omega, \omega') / \text{pdf})
Photon Mapping

A two-pass algorithm:

- Pass 1: Trace virtual photons from the light source, scatter at surfaces, and cache
- Pass 2: Ray trace the scene and use the photons to compute indirect illumination
For each shading point:

- Find the k closest photons
- Approx. radiance using density of photons
For each shading point:

- Find the k closest photons
- Approx. radiance using density of photons
The Radiance Estimate

\[ L_r(x, \vec{\omega}) \approx \sum_{p=1}^{k-1} f_r(x, \vec{\omega}_p, \vec{\omega}) \frac{\Phi_p}{A_k} \]
The Radiance Estimate

\[ L_r(x, \omega) \approx \sum_{p=1}^{k-1} f_r(x, \omega_p, \bar{\omega}) \frac{\Phi_p}{\pi r^2 k} \]
The Radiance Estimate

\[ L_r(x, \vec{\omega}) \approx \sum_{p=1}^{k-1} f_r(x, \vec{\omega}_p, \vec{\omega}) \Phi_p K_{2D}(r_p, r_k) \]
The Photon Map Data Structure

Requirements:
- Compact (we want many photons)
- Fast nearest neighbor search

KD-tree
Global Illumination

100000 photons / 50 photons in radiance estimate
Global Illumination

500000 photons / 500 photons in radiance estimate

Tuesday 22 April 14
Path Tracing
Practical Photon Mapping

Split illumination
Caustics

- Separate, higher quality photon map for caustics
  - Only stores LS^D paths
  - Many photons shot directly at specular objects
Density estimation is blotchy

Use final gather
Global Illumination

500000 photons / 500 photons in radiance estimate
Improved Photon Mapping

final gather + global photon map (200000) + caustic photon map (50000)
Path Tracing
Next...

- Improve / extend / generalize
- Progressive Photon Mapping