Rendering Equation
Biased MC Methods

Image Synthesis
Torsten Möller
Reading

• Chapter 16.4 of “Physically Based Rendering” by Pharr & Humphreys

• Chapter 19, 20 in “Principles of Digital Image Synthesis,” by A. Glassner

• “Realistic Image Synthesis Using Photon Mapping,” by H. W. Jensen

© Machiraju/Möller
Biased vs. Unbiased Algorithm

- Unbiased algorithms:
  - Noise very apparent OR we need tons of rays
  - Variance (error) decreases in a predictable way

- Biased approaches
  - Irradiance caching + photon mapping
  - Smooth results
  - Cheap (only little more computing time than Whitted ray-tracing)
Biased vs. Unbiased Algorithm

• If we find little noise in the image:
  – For unbiased methods
    • can be fairly sure that image is correct
  – For biased methods
    • don’t have this assurance
    • Increasing the sampling rate might not get rid of existing artifacts
Irradiance Caching

- Introduced by Greg Ward 1988
- Implemented in RADIANCE
  - Public-domain software
- Exploits smoothness of irradiance
  - Cache and interpolate irradiance estimates
Irradiance Caching

- Indirect lighting changes rather slowly
- Compute indirect light at sparse set of samples
- Interpolate neighboring values from this set of samples
Irradiance Caching

A. How to come up with such a (sparse) set of samples?

B. How to store these samples?

C. How to interpolate from neighbors?
A) Set of Samples

- Find them adaptively
- If there are no good nearby samples then compute a new irradiance sample
- Store irradiance (radiance is directionally dependent; i.e. expensive to store):
  \[ E(p,n) = \int_{H^2(n)} L_i(p,\omega_i) |\cos \theta_i| d\omega_i \]
- we have: \[ L_o(p,\omega_o) = \int_{H^2(n)} f_r(p,\omega_o,\omega_i) L_i(p,\omega_i) |\cos \theta_i| d\omega_i \]
- Assuming Lambertian-type surfaces (f=const.) \[ L_o(p,\omega_o) = c \cdot E(p,n) \]
A) Set of Samples

• More glossy surfaces:

\[ L_o(p, \omega_o) \approx \left( \int_{H^2(n)} f_r(p, \omega_o, \omega_i) d\omega_i \right) \left( \int_{H^2(n)} L_i(p, \omega_i) |\cos \theta_i| d\omega_i \right) \]

\[ = \frac{1}{2} \rho_{hd}(\omega_o) \cdot E(p, n) \]

• \( \rho_{hd}(\omega_o) \) hemispherical-directional reflectance
A) Set of Samples

Irradiance caching  Path Tracing

Same Computation Time
A) Set of Samples

Irradiance caching
Irradiance samples used
A) Set of Samples

- High error for specular surface
- Specular BSDF => Whitted integrator
- Diffuse BSDF => irradiance caching
  - Interpolation from known points
  - A path tracing step for new points

\[
E(p,n) = \frac{1}{N} \sum_{j} \frac{L_i(p,\omega_j) \left| \cos \theta_j \right|}{p(\omega_j)}
\]

- Use cos-weighted distribution: \( p(\omega) = \frac{\cos \theta}{\pi} \)

\[
E(p,n) = \frac{\pi}{N} \sum_{j} L_i(p,\omega_j)
\]

© Machiraju/Möller
B) Storing Samples

- **Octree data structure**
  - Each node stores samples that influence this node (each point has a radius of influence!)

- **Radius of influence**
  - determined by harmonic mean $\frac{N}{\sum_{i}^{N} \frac{1}{d_i}}$
  - $d_i$ is the distance that the $i$th ray (used for estimating the irradiance) traveled before intersecting an object
  - Computed during path tracing
C) Interpolation from neighbors

- Simply weighted sum: \[ E = \frac{\sum_i w_i E_i}{\sum_i w_i} \]
- With weights:
  \[ w_i = \left( \frac{d_{\text{max}} \cdot (N \cdot N') - d}{d_{\text{max}} \cdot (N \cdot N')} \right)^2 \]
- Skip samples
  - Normals too different
  - Too far away
  - If in front
Images from “Radiance”
Images from “Radiance”
Caching Techniques

• Irradiance caching:
  – Compute irradiance at selected points and interpolate.

• Photon mapping:
  – Trace “photons” from the lights and store them in a photon map, that can be used during rendering.
Photon Mapping

• Basic idea: Density estimation with a discrete density of photons

• 2-step algorithm
  – Photon trace:
    • Simulates the transport of individual photons
    • Photons emitted from source
    • Photons deposited on surfaces
    • Photons reflected from surfaces to other surfaces
  – Rendering
    • Photons collected by rendering
First Pass - Photon Trace

For 100 photons emitted from 100W source, each photon initially carries 1W.

Propagate this radiant flux through scene using MC methods.

© Machiraju/Möller
Estimating incident flux

- At any patch of surface, we can estimate the incident flux:
  
  Just average the contributions of all the photons that hit the patch.

© Machiraju/Möller
A Photon

- For each surface interaction, we store:

```c
struct photon {
    float x, y, z; // position
    char power[4]; // power (R, G, B, E)
    char phi, theta; // incident direction
    short flag;
}
```
Photon Storage

• Store this information about surface interactions in photon map (kd-tree)
• Photon storage is decoupled from geometry
• Estimate flux incident at a surface point based on nearby photons.
Radiance Estimate

- Expand ball until it contains some reasonable number of photons.
- Use intersection with plane to estimate area of surface patch.
Radiance Estimate

• Recall the reflected radiance equation
  \[
  L_o(p, \omega_o) = \int_{S^2} f_r(p, \omega_i, \omega_o)L_i(p, \omega_i)|\cos \theta_i|d\omega_i
  \]

• Convert incident radiance into incident flux
  \[
  L_i(p, \omega_i) = \frac{d^2\Phi_i(p, \omega_i)}{|\cos \theta_i|d\omega_i dA_i}
  \]

• Reflected radiance in terms of incident flux
  \[
  L_o(p, \omega_o) = \int_{S^2} f_r(p, \omega_i, \omega_o) \frac{d^2\Phi_i(p, \omega_i)}{dA_i} dA_i
  \]

• Numerically
  \[
  L_o(p, \omega_o) \approx \frac{1}{\Delta A} \sum_{p=1}^{n} f_r(p, \omega_p, \omega_o)\Delta \Phi_p(p, \omega_p)
  \]

© Machiraju/Möller
Caustic from a Glass Sphere

• Photon Mapping: 10000 photons / 50 photons in radiance estimate
Caustic from a Glass Sphere

- Path Tracing: 1000 paths/pixel
Sphereflake Caustic
Reflection Inside A Metal Ring

- 50000 photons / 50 photons in radiance estimate
Caustics On Glossy Surfaces

- 340000 photons / 100 photons in radiance estimate
HDR environment illumination

- Using lightprobe from www.debevec.org
Cognac Glass
Cube Caustic
Global Illumination

• 100000 photons / 50 photons in radiance estimate
Global Illumination

- 500000 photons / 500 photons in radiance estimate
Fast estimate

- 200 photons / 50 photons in radiance estimate
Indirect illumination

- 10000 photons / 500 photons in radiance estimate
Global Illumination

- global photon map vs. caustics photon map
Photon tracing

- Photon emission
- Photon scattering
- Photon storing
Photon emission

• Given $\Phi$ Watt lightbulb.
• Emit $N$ photons.
• Each photon has the power $\Phi/N$ Watt
• Photon power depends on the number of emitted photons. Not on the number of photons in the photon map.

© Machiraju/Möller
What is a photon?

• Flux (power) - not radiance!
• Collection of physical photons
  – A fraction of the light source power
  – Several wavelengths combined into one entity
Diffuse point light

• Generate random direction
• Emit photon in that direction

// Find random direction
do {
    x = 2.0*random()-1.0;
    y = 2.0*random()-1.0;
    z = 2.0*random()-1.0;
} while ( (x*x + y*y + z*z) > 1.0 );
Example: Diffuse square light

- Generate random position $p$ on square
- Generate diffuse direction $d$
- Emit photon from $p$ in direction $d$

```cpp
// Generate diffuse direction
u = random();
v = 2*π*random();
d = vector(cos(v)√u, sin(v)√u, √1−u);
```
Other Sources

- Random directions and importance sampling (e.g., diffuse) within volumes of emission (bounding volumes)
- Projection Maps – Cells in scene which indicate if photons map or not
Surface interactions

• The photon is
  – Stored (at diffuse surfaces) and
  – Absorbed (A) or
  – Reflected (R) or
  – Transmitted (T)

\[ A + R + T = 1.0 \]
Photon scattering

• The simple way:
  – Given incoming photon with power $\Phi_p$
  – Reflect photon with the power $R^*\Phi_p$
  – Transmit photon with the power $T^*\Phi_p$
Photon scattering

• The simple way:
  – Given incoming photon with power $\Phi_p$
  – Reflect photon with the power $R*\Phi_p$
  – Transmit photon with the power $T*\Phi_p$

• Risk: Too many low-powered photons - wasteful!

• When do we stop (systematic bias)?

• Photons with similar power is a good thing.
Russian Roulette

• Probability of termination: $q$

\[
E\{X\} = q \cdot 0 + (1 - q) \frac{E\{X\}}{1 - q} = E\{X\}
\]

• Terminate un-important photons and still get the correct result
Russian Roulette Example

- Surface reflectance: \( R = 0.5 \)
- Incoming photon: \( \Phi_p = 2W \)

\[
\begin{align*}
  r &= \text{random}(); \\
  \text{if } ( r < 0.5 ) &\quad \text{reflect photon with power 2 W} \\
  \text{else} &\quad \text{photon is absorbed}
\end{align*}
\]
Russian Roulette Intuition

• Surface reflectance: $R = 0.5$
• 200 incoming photons with power: $\Phi_p = 2\text{W}$
• Reflect 100 photons with power 2W
• instead of 200 photons with power 1W
Russian Roulette

- Very important!
- Use to eliminate un-important photons
- Gives photons with similar power
Storing Photons

• data structure for storing and searching
• For uniform distribution – cubes are best
• For non-uniform – kd-trees, Voronoi etc.
• kd-trees, perhaps best
Storing Photons

- Use kd-tree – a sequence of axis-aligned partitions
  - 2-D partitions are lines
  - 3-D partitions are planes
- Axis of partitions alternates with depth of the tree
- Average access time - $O(\log n)$
- Worst case $O(n)$ when tree is skewed
- Need to maintain a balanced tree
- Balancing $O(n \log n)$
- $k$ nearest neighbors in $O(k + \log n)$
Balancing kd-tree

```c
kdtree *balance( points ) {
    Find the cube surrounding the points
    Select dimension \textit{dim} in which the cube is largest
    Find median of the points in \textit{dim}
    \( s1 = \text{all points below median} \)
    \( s2 = \text{all points above median} \)
    \( \text{node} = \text{median} \)
    \( \text{node.left} = \text{balance}(s1) \)
    \( \text{node.right} = \text{balance}(s2) \)
    return node
}
```
Multiple Photon Maps

- Interpolation from the map causes blurriness
- For specular interactions
  Whitted Ray-Tracing
- Otherwise divide photon map into
  - direct light map $L_{i,d}$
  - Indirect light map $L_{i,i}$
  - Caustic map $L_{i,c}$
Multiple Photon Maps

- Direct
  - Emission
  - Specular
  - Nonspecular

- Caustic
  - Specular
  - Nonspecular

- Indirect
  - Nonspecular
  - Specular / Nonspecular
Photon Mapping: Rendering

- Classification of photons in Photon Map

Figure 1: The photons in the global photon map are classified to optimize the rendering of shadows
Multiple Photon Maps

- Direct light map $L_{i,d}$
- Indirect light map $L_{i,i}$
- Caustic map $L_{i,c}$

$$L_o(p,\omega_o) = \int_{S^2} f_r(p,\omega_i,\omega_o)L_i(p,\omega_i)|\cos \theta_i|d\omega_i$$

$$= \int_{S^2} f_\Delta(p,\omega_i,\omega_o)L_i(p,\omega_i)|\cos \theta_i|d\omega_i$$

$$+ \int_{S^2} f_\Delta(p,\omega_i,\omega_o)(L_{i,d}(p,\omega_i) + L_{i,i}(p,\omega_i) + L_{i,c}(p,\omega_i))|\cos \theta_i|d\omega_i$$
Another example

Direct Illumination + Indirect Illumination = Specular Part
Another example (cont’d)
Features

• Photon tracing is unbiased
  – Radiance estimate is biased but consistent
  – The reconstruction error is local

• Illumination representation is decoupled from the geometry
Box

- 200000 global photons, 50000 caustic photons
Box: Global Photons

- 200000 global photons
Fractal Box

- 200000 global photons, 50000 caustic photons
Cornell Box
Indirect Illumination
Little Matterhorn

© Machiraju/Möller
Mies house (swimmingpool)
Mies house (3pm)
Mies house (6pm)
Final Gathering

• Used for higher quality indirect light
• Use BSDF-based distribution of rays at first hit-point in the scene in order to gather the light from direct, indirect and caustic photon maps of places intersected by these rays!
Direct Light only
Photon Map = no Final Gathering
Photon Map = Final Gathering
Recap

- Decouple representation of illumination from geometry
- Photon Map –
  - Illumination as points in a global data structure
  - Cache of light paths
- Estimate illumination based on density
  - Error is of low frequency
  - Not high frequency noise
- Biased method! Average expected value is not correct!
- Consistent – technique will converge as more points/photons are added
- Photon Mapping, Photon Tracing, Photon Map