Biased Monte Carlo Ray Tracing: Filtering, Irradiance Caching and Photon Mapping

Dr. Henrik Wann Jensen
Stanford University
May 24, 2001
Unbiased and consistent Monte Carlo methods

Unbiased estimator:

$$E\{X\} = \int \ldots$$

Consistent estimator:

$$\lim_{N \to \infty} E\{X\} \to \int \ldots$$
Unbiased and consistent: A very simple example

Unbiased estimator:

\[
\frac{1}{N} \sum_{i=1}^{N} f(\xi_i)
\]

Consistent estimator:

\[
\frac{1}{N + 1} \sum_{i=1}^{N} f(\xi_i)
\]
Path tracing (unbiased)

10 paths/pixel

Biased Monte Carlo ray tracing: filtering, irradiance caching and photon mapping
Path tracing (unbiased)

10 paths/pixel
Path tracing (unbiased)

100 paths/pixel
How can we remove this noise?)
How can we remove this noise?)

- More samples (slow convergence, $\sigma \propto 1/\sqrt{N}$)
How can we remove this noise?

- More samples (slow convergence, $\sigma \propto 1/\sqrt{N}$)
- Better sampling (stratified, importance, qmc etc.)
How can we remove this noise?)

- More samples (slow convergence, $\sigma \propto 1/\sqrt{N}$)
- Better sampling (stratified, importance, qmc etc.)
- Adaptive sampling
How can we remove this noise?

- More samples (slow convergence, $\sigma \propto 1/\sqrt{N}$)
- Better sampling (stratified, importance, qmc etc.)
- Adaptive sampling
- Filtering
How can we remove this noise?

- More samples (slow convergence, $\sigma \propto 1/\sqrt{N}$)
- Better sampling (stratified, importance, qmc etc.)
- Adaptive sampling
- Filtering
- Caching and interpolation
Stratified sampling

10 paths/pixel
Quasi Monte-Carlo (Halton sequence)

10 paths/pixel
Fixed (Random) Sequence

10 paths/pixel

Biased Monte Carlo ray tracing: filtering, irradiance caching and photon mapping
Filtering: idea

- Noise is high frequency
Filtering: idea

- Noise is high frequency
- Remove high frequency content
Unfiltered image

10 paths/pixel
3x3 lowpass filter

10 paths/pixel
3x3 median filter

10 paths/pixel
Energy preserving filters
Energy preserving filters

- Distribute noisy energy over several pixels
Energy preserving filters

- Distribute noisy energy over several pixels
- Adaptive filter width
- Diffusion style filters
- Splatting style filters
Problems with filtering

• Everything is filtered (blurred)
  ★ Textures
  ★ Highlights
  ★ Caustics
  ★ . . .
Problems with filtering

- Everything is filtered (blurred)
  - Textures
  - Highlights
  - Caustics
  - ...

Solution: Try to filter the noisy part of the illumination
Caching Techniques
Caching Techniques

Irradiance caching: Compute irradiance at selected points and interpolate.

Photon mapping: Density estimation and importance sampling using a precomputed flux representation.
Box: direct illumination
Box: global illumination
Box: indirect irradiance
Irradiance caching: idea


Idea: Irradiance changes slowly → interpolate.
Irradiance sampling

\[ E(x) = \int_{2\pi} L'(x, \omega') \cos \theta \, d\omega' \]
Irradiance sampling

\[ E(x) = \int_{2\pi} L'(x, \omega') \cos \theta \, d\omega' \]
\[ = \int_{0}^{2\pi} \int_{0}^{\pi/2} L'(x, \theta, \phi) \cos \theta \sin \theta \, d\theta \, d\phi \]
Irradiance sampling

\[ E(x) = \int_{2\pi} L'(x, \omega') \cos \theta \, d\omega' \]

\[ = \int_0^{2\pi} \int_0^{\pi/2} L'(x, \theta, \phi) \cos \theta \sin \theta \, d\theta \, d\phi \]

\[ \approx \frac{\pi}{TP} \sum_{t=1}^T \sum_{p=1}^P L'(\theta_t, \phi_p) \]

\[ \theta_t = \sin^{-1}\left(\sqrt{\frac{t-\xi}{T}}\right) \quad \text{and} \quad \phi_p = 2\pi \frac{p-\psi}{P} \]

Biased Monte Carlo ray tracing: filtering, irradiance caching and photon mapping
Irradiance change

\[ \epsilon(x) \leq \left| \frac{\partial E}{\partial x}(x - x_0) + \frac{\partial E}{\partial \theta}(\theta - \theta_0) \right| \]

\( \epsilon(x) \) \hspace{1cm} \text{position} \hspace{1cm} \text{rotation}
Irradiance change

\[ \epsilon(x) \leq \left| \frac{\partial E}{\partial x}(x - x_0) + \frac{\partial E}{\partial \theta}(\theta - \theta_0) \right| \]

\[ \leq E_0 \left( \frac{4 ||x - x_0||}{\pi x_{avg}} + \sqrt{2 - 2\vec{N}(x) \cdot \vec{N}(x_0)} \right) \]
Irradiance interpolation

\[ w(x) = \frac{1}{\epsilon(x)} \approx \frac{1}{\|x-x_0\|_{x_{avg}}} + \sqrt{1 - \vec{N}(x) \cdot \vec{N}(x_0)} \]

\[ E_i(x) = \frac{\sum_i w_i(x)E(x_i)}{\sum_i w_i(x)} \]
Irradiance caching algorithm

Find all irradiance samples with \( w(x) > q \)

if (samples found)
   interpolate
else
   compute new irradiance sample
Box: irradiance gradients

1000 sample rays, w > 10
Box: irradiance cache positions

1000 sample rays, $w > 10$
Box: irradiance gradients

1000 sample rays, $w > 20$
Box: irradiance cache positions

1000 sample rays, \( w > 20 \)
Box: irradiance gradients

5000 sample rays, $w > 10$
Box: irradiance cache positions

5000 sample rays, \( w > 10 \)
Photon Mapping

Two-pass method:

Pass 1 : Build a photon map using photon tracing

Pass 2 : Render the image using the photon map
A simple test scene
Building the Photon Map: Photon Tracing
Photons
The photon map datastructure

The photons are stored in a left balanced kd-tree

```c
struct photon = {
    float position[3];
    rgbe power; // power packed as 4 bytes
    char phi, theta; // incoming direction
    short flags;
};
```
Radiance estimate

\[ L(x, \vec{\omega}) = \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) L'(x, \vec{\omega}') \cos \theta' \, d\omega \]
Radiance estimate

\[ L(x, \bar{\omega}) = \int_{\Omega} f_r(x, \bar{\omega}', \bar{\omega}) L'(x, \bar{\omega}') \cos \theta' \, d\omega = \int_{\Omega} f_r(x, \bar{\omega}', \bar{\omega}) \frac{d\Phi^2(x, \bar{\omega}')}{d\omega} \cos \theta' \, dA \cos \theta' \, d\omega \]
Radiance estimate

\[
L(x, \omega) = \int_{\Omega} f_r(x, \omega', \omega) L'(x, \omega') \cos \theta' \, d\omega \\
= \int_{\Omega} f_r(x, \omega', \omega) \frac{d\Phi^2(x, \omega')}{d\omega \cos \theta' \, dA} \cos \theta' \, d\omega \\
= \int_{\Omega} f_r(x, \omega', \omega) \frac{d\Phi^2(x, \omega')}{dA}
\]
Radiance estimate

\[ L(x, \vec{\omega}) = \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) L'(x, \vec{\omega}') \cos \theta' \, d\omega \]

\[ = \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) \frac{d\Phi^2(x, \vec{\omega}')}{d\omega \cos \theta' \, dA} \cos \theta' \, d\omega \]

\[ = \int_{\Omega} f_r(x, \vec{\omega}', \vec{\omega}) \frac{d\Phi^2(x, \vec{\omega}')}{dA} \]

\[ \approx \sum_{p=1}^{n} f_r(x, \vec{\omega}'_p, \vec{\omega}) \frac{\Delta \Phi_p(x, \vec{\omega}'_p)}{\pi r^2} \]
Radiance estimate
Reflection inside a metal ring

50000 photons / 50 photons in radiance estimate
Caustics on glossy surfaces

340000 photons / \approx 100\ photons\ in\ radiance\ estimate
Cognac glass
Cube caustic
Caustic from a glass sphere

10000 photons / 50 photons in radiance estimate
Caustic from a glass sphere
Path tracing

1000 paths/pixel
Caustic from a glass sphere in Grace Cathedral

Using lightprobe from www.debevec.org
Direct visualization of the radiance estimate

100000 photons / 50 photons in radiance estimate
Direct visualization of the radiance estimate

500000 photons / 500 photons in radiance estimate
Fast estimate

200 photons / 50 photons in radiance estimate
Only use photons for indirect irradiance

10000 photons / 500 photons in radiance estimate
Two photon maps

global photon map

caustics photon map
Rendering
Rendering: direct illumination
Rendering: specular reflection
Rendering: caustics
Rendering: indirect illumination

Biased Monte Carlo ray tracing: filtering, irradiance caching and photon mapping
Two-pass method

\[ \text{Radiance} = \text{direct illumination} + \]
\[ \quad \text{specular reflection/transmission} + \]
\[ \quad \text{caustics} + \]
\[ \quad \text{soft indirect irradiance} \]
Rendering Equation Solution

\[
L_r(x, \vec{\omega}) = \int_{\Omega_x} f_r(x, \vec{\omega}', \vec{\omega}) L_i(x, \vec{\omega}') \cos \theta_i \, d\omega'_i \\
= \int_{\Omega_x} f_r(x, \vec{\omega}', \vec{\omega}) L_{i,l}(x, \vec{\omega}') \cos \theta_i \, d\omega'_i + \\
\int_{\Omega_x} f_{r,s}(x, \vec{\omega}', \vec{\omega}) (L_{i,c}(x, \vec{\omega}') + L_{i,d}(x, \vec{\omega}')) \cos \theta_i \, d\omega'_i + \\
\int_{\Omega_x} f_{r,d}(x, \vec{\omega}', \vec{\omega}) L_{i,c}(x, \vec{\omega}') \cos \theta_i \, d\omega'_i + \\
\int_{\Omega_x} f_{r,d}(x, \vec{\omega}', \vec{\omega}) L_{i,d}(x, \vec{\omega}') \cos \theta_i \, d\omega'_i.
\]
200,000 global photons, 50,000 caustic photons
Fractal box

200000 global photons, 50000 caustic photons
Sphereflake caustic
Little Matterhorn
Mies house (swimmingpool)
Mies house (3pm)
Mies house (6pm)
Mies house (7pm)
Participating media
Participating media: photon tracing
The volume photon map
The volume radiance estimate

\[(\vec{\omega} \cdot \nabla) L(x, \vec{\omega}) = \sum_{p=1}^{n} p(x, \vec{\omega}'_p, \vec{\omega}) \frac{\Delta \Phi_p(x, \vec{\omega}'_p)}{4 \pi r^3}\]
Rendering participating media
Volume caustic
Rising smoke
Rising smoke
Subsurface scattering

- Skin
- Marble
- Actually most materials
Subsurface scattering
David (subsurface scattering)
David (subsurface scattering)
Diana the Huntress
Diana the Huntress
Diana the Huntress
Diana the Huntress
Diana the Huntress: no subsurface scattering
Diana the Huntress: subsurface scattering
More information

http://graphics.stanford.edu/~henrik/

henrik@graphics.stanford.edu