Global Illumination Using Photon Maps

A summary of the paper by Henrik W. Jensen [Jen96]

Martin Estgren <mares480@student.liu.se>
Erik S. V. Jansson <erija578@student.liu.se>

Advanced Global Illumination and Rendering
at ITN Linköping University (LiTH), Sweden

October 31, 2017

Outline

- Motivation
- Global Illumination
  - Rendering Equation
  - Radiosity
  - Whitted Raytracing
  - Path Tracing
- Photon Mapping
  - Photon Tracing
  - Radiance Estimate
  - Photon Collection
- Summary
- Further Studies

Motivation

- We want a global illumination scheme that achieves photorealism as $N \to \infty$
- It should also preferably be noise-free!
- Should also be physically sound model
- Preferably also relat. fast to compute!

- We'll show a couple of schemes trying to achieve these, where they fall short, and how *photon mapping* can be used.

Global Illumination

Rendering Equation [Kaj86]

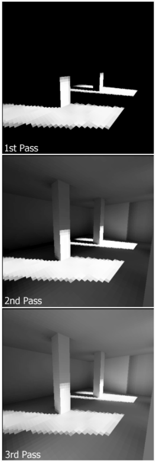
$$L_o(\vec{x},\hat{\omega}_o) = L_e(\vec{x},\hat{\omega}_o) + \overbrace{\int_\Omega L_i(\vec{x},\hat{\omega}_i) f_r(\vec{x},\hat{\omega}_i,\hat{\omega}_o)(\hat{n}_x \cdot \hat{\omega}_i)\, d\hat{\omega}_i}^{\textit{reflected radiance } L_r \text{ from } \Omega \text{ to } \vec{x} \text{ toward } \hat{\omega}_o}$$

- $L_o(\vec{x},\hat{\omega}_o)$: total *outgoing radiance* at point $\vec{x}$ towards a $\hat{\omega}_o$.
- $L_e(\vec{x},\hat{\omega}_o)$: *emitted radiance* contribution from $\vec{x}$ toward $\hat{\omega}_o$.
- Ω: hemisphere around the point $\vec{x}$ with normal $\hat{n}_x$ of $d\hat{\omega}_i$'s.
- $L_i(\vec{x},\hat{\omega}_i)$: *incoming radiance* contributions fr. $\hat{\omega}_i$ towards $\vec{x}$.
- $f_r(\vec{x},\hat{\omega}_i,\hat{\omega}_o)$: surface *reflectance properties* at $\vec{x}$, an BRDF.

Assumes all surfaces are Lambertian reflectors: $f_r = \rho/\pi$ of discrete size.

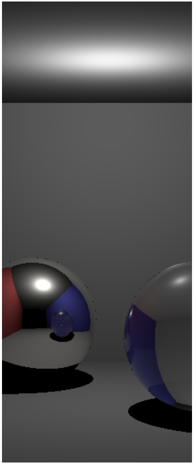
$$B_i = E_i + \rho_i \sum_{j=1}^{n} F_{ij} B_j$$

- + Noise free, iterative progression
- + Viewport independent (baking!)
- + Accurate for Lambertian surface
- – No specular or glossy reflections
- – Complexity scales $\propto$ to triangles



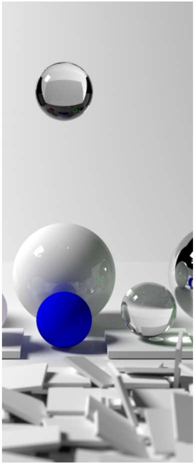
Global Illumination
Whitted Raytracing [Whi79]

- $\sim$ Viewport dependent (no baking)
- $-$ No soft shadows, $\rightarrow$ point lights
- $+$ Enables fully specular reflections
- $-$ Uses local model for diffuse surf.
- $+$ OK to parallelize and implement



Global Illumination
Path Tracing, MC Raytracer

+ Models almost all light transport
+ Embarrassingly parallel algorithm
– High-freq. noise if undersampled
– Doesn't consider vol. interaction
– Unfeasible for "detailed caustics"

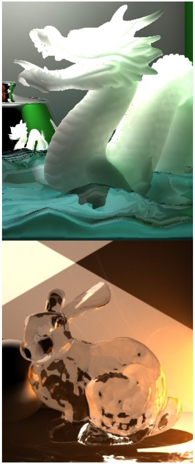


Photon Mapping

Efficient ray tracing extension that features faster *caustics*, and, *sub-surface scattering*. Trick: send *"photons"* from a light source.

Algorithm is divided into two major passes:

- Photon tracing: emit photons carrying flux $d\Phi$ from light sources towards our scene and stores hits on *photon maps*.
- Photon collection: estimate irradiance at $\vec{x}$ by integrating over *photon maps*. Either *accurate*, or *approximate* mode.



Photon Tracing

Photons are emitted from light source and traced similar to path tracing and terminated on diffuse surfaces.

Caustics $\approx L_{i,c}(\vec{x}, \hat{\omega}_i)$

- Photons are only emitted towards specular objects
- Used to directly visualize caustics

Global $\approx L_{i,d}(\vec{x}, \hat{\omega}_i)$

- Photons are emitted uniformly over the hemisphere
- Only used as a data structure

Surface interaction is determined by russian roulette.

Assume a uniform real distribution $\xi \in [0, 1]$ and the surface coefficients r and t:

- $\xi \in [0, r]$ - Reflection of photon
- $\xi \in [r, r + t]$ - Transmission of photon
- $\xi \in [r + t, 1]$ - Absorption of photon

Radiance Estimate

The reflected radiance from a point $\vec{x}$ can be estimated as:

$$L_r(\vec{x}, \hat{\omega}_o) \approx \sum_{p=1}^{n} f_r(\vec{x}, \hat{\omega}_{i,p}, \hat{\omega}_o) \frac{\Delta\Phi_p(\vec{x}, \hat{\omega}_{i,p})}{\pi r^2}$$

Breaking down the estimate:

1. Locate the n nearest photons p with *flux* $\Delta\Phi_p$ around $\vec{x}$
2. Approximate the area containing the photons as πr^2 (circle)
3. For each photon: Multiply the *BRDF* $f_r(\vec{x}, \hat{\omega}_{i,p}, \hat{\omega}_o)$ with the *flux* divided by the approximate containment area

Possible optimization: By using a fixed bounding sphere around $\vec{x}$ instead of the n nearest photons.

Photon Collection

After we have our photon maps of the scene, we use distribution raytracing to compute each pixel's average radiance by sampling estimates from the scene: $L_o(\vec{x}, \hat{\omega}_o) = L_e(\vec{x}, \hat{\omega}_o) + L_r(\vec{x}, \hat{\omega}_o)$. In photon mapping we split $L_r(\vec{x}, \hat{\omega}_o)$ into 4 radiance contributions:

$$L_r(\vec{x}, \hat{\omega}_o) = \int_{\Omega} L_i(\vec{x}, \hat{\omega}_i) f_r(\vec{x}, \hat{\omega}_i, \hat{\omega}_o)(\hat{n}_x \cdot \hat{\omega}_i)\, d\hat{\omega}_i =$$
$$L_{r,l}(\vec{x}, \hat{\omega}_o) + L_{r,s}(\vec{x}, \hat{\omega}_o) + L_{r,c}(\vec{x}, \hat{\omega}_o) + L_{r,d}(\vec{x}, \hat{\omega}_o)$$

Each of these are evaluated either *accurately* or *approximatively* :

- Accurate evaluation: when $\vec{x}$ is seen by an eye directly/close
- Approximate evaluation: a importance was reflected diffusely

Photon Collection
Direct Illumination

Represents the reflected radiance at $\vec{x}$ towards $\hat{\omega}_o$ which originated directly from $\hat{\omega}_i$, the sources of light found around a hemisphere Ω.

$$L_{r,l}(\vec{x}, \hat{\omega}_o) = \int_\Omega L_{i,l}(\vec{x}, \hat{\omega}_i) f_{r,d}(\vec{x}, \hat{\omega}_i, \hat{\omega}_o)(\hat{n}_x \cdot \hat{\omega}_i)\, d\hat{\omega}_i$$

- Accurate evaluation: we would have to launch *shadow rays* in the same way as Monte Carlo raytracing. Sample if area light.
- Approximate evaluation: the radiance estimate obtained from our *global photon map*, sum estimated photon flux around $\vec{x}$.

Photon Collection
Specular and Glossy Reflections

Specular and glossy reflections are handled using the sum over the incoming caustic and diffuse radiance with the specular BRDF $f_{r,s}$. No approximated variant is needed, we only have a single direction.

$$L_{r,s}(\vec{x},\hat{\omega}_o) = \int_\Omega L_{i,c}(\vec{x},\hat{\omega}_i) f_{r,s}(\vec{x},\hat{\omega}_i,\hat{\omega}_o)(\hat{n}_x\cdot\hat{\omega}_i)\,d\hat{\omega}_i + \int_\Omega L_{i,d}(\vec{x},\hat{\omega}_i) f_{r,s}(\vec{x},\hat{\omega}_i,\hat{\omega}_o)(\hat{n}_x\cdot\hat{\omega}_i)\,d\hat{\omega}_i$$

- Accurate evaluation: we use Monte Carlo raytracing and $f_{r,s}$, distributed importance sampling for reflection & transmission.

Contributions from caustics are never evaluated using Monte Carlo raytracing, since it's very expensive. We use photon maps instead. $L_{i,c}(\vec{x}, \hat{\omega}_o)$ is indirect light via specular reflection, or, transmission.

$$L_{r,c}(\vec{x}, \hat{\omega}_o) = \int_\Omega L_{i,c}(\vec{x}, \hat{\omega}_i) f_{r,d}(\vec{x}, \hat{\omega}_i, \hat{\omega}_o)(\hat{n}_x \cdot \hat{\omega}_i)\, d\hat{\omega}_i$$

- Accurate evaluation: we use radiance estimate of the *caustics photon map*. This is why it needs to be a high-resolution map.
- Approximate evaluation: again, we use the global photon map.

Photon Collection
Indirect Diffuse Reflections

Finally, for contributions arriving at $\vec{x}$ which have bounced around diffusely at least once we integrate over $L_{i,d}(\vec{x}, \hat{\omega}_i)$. This gives the visual effects commonly known as "color bleeding", as in radiosity.

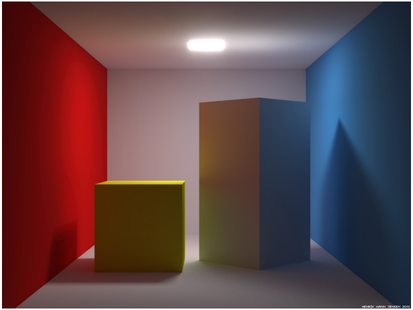
$$L_{r,d}(\vec{x}, \hat{\omega}_o) = \int_{\Omega} L_{i,d}(\vec{x}, \hat{\omega}_i) f_{r,d}(\vec{x}, \hat{\omega}_i, \hat{\omega}_o)(\hat{n}_x \cdot \hat{\omega}_i)\, d\hat{\omega}_i$$

- Accurate evaluation: we use Monte Carlo raytracing again for this, and an optimzed sampling distribution according to $f_{r,d}$.
- Approximate evaluation: will usually contribute quite little to the end-results, so we again use the global radiance estimate.

Photon Mapping
Putting It All Together

- We emit photons from the sources of light, bounce these around the scene while recording flux hitting the surfaces.
  - These photon maps are usually stored by using a k-d tree.
- We can estimate the radiance arriving at any point using these photon maps by picking the k-NN photons' flux $\Delta\Phi$.
- Finally, we can more efficietly run Monte Carlo raytracing by adaptively choosing an *accurate* or *approximate* mode:
  - Accurate: when we see surfaces directly, use path tracing.
  - Approximate: use the estimate given by our photon maps.

Photon Mapping
Image Samples in [Jen96]

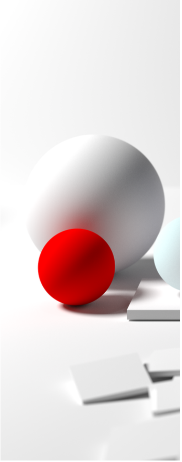


Summary

In this presentation we have explained:

- The desirable properties of accurate global illumination algorithms
- Where the existing models fall short
- How photon mapping improves path tracing by using a radiance estimate
  - Photon tracing, photon collection

We'll now discuss some improvements.



Further Studies

- Improved radiance estimate: we add a filter based on the distance which re-weight the contributions from photons.
- Volume photon map: models interactions in participating media. We need a new radiance estimate and a “BSDF”.
- Photon splatting: instead of storing a photon individually, we accumulate the radiance in a texture with all the hits.

Any Questions?

Bibliography

Cindy M Goral, Kenneth E Torrance, Donald P Greenberg, and Bennett Battaile.
Modeling the interaction of light between diffuse surfaces.
In *ACM SIGGRAPH Computer Graphics*, volume 18, pages 213–222. ACM, 1984.

Henrik Jensen, Per Christensen, Toshiaki Kato, and Frank S.
A practical guide to global illumination using photon mapping.
SIGGRAPH 2002 Course Notes CD-ROM, 2002.

Henrik Wann Jensen.
Global illumination using photon maps.
Rendering Techniques, 96:21–30, 1996.

James T Kajiya.
The rendering equation.
In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.

Chun Kang, Lu Wang, Yan Xu, and Xiang-xu Meng.
A survey of state-of-the-art photon mapping and future work.
Frontiers of IT & Electronic Engineering, 17(3):185–199, 2016.

Turner Whitted.
An improved illumination model for shaded display.
In *ACM SIGGRAPH Computer Graphics*, volume 13, page 14. ACM, 1979.