COMS 30115

Stochastic Raytracing & Photon mapping

Carl Henrik Ek - carlhenrik.ek@bristol.ac.uk

March 23rd, 2018

http://www.carlhenrik.com

Introduction

Last Time

- Clarification of transport problem
- Area formulation of Rendering Equation
- Radiosity

Today

- Approximate integration
- Monte Carlo Methods
- Photon mapping

Material

Stochastic Pathtracing

- Thesis Appendix of Wojciech Jarosz
- Equation Compendium
- Global Illumination Resources

Photon Mapping

- Henrik Jensen Siggraph Tutorial

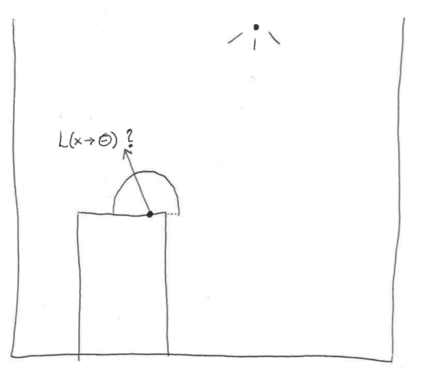
Stochastic Raytracing

Transport of Light

$$L(x \to \Theta) = L_e(x \to \Theta) + \mathcal{T}(L(x \to \Theta))$$

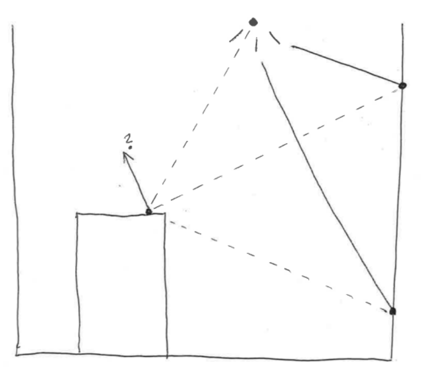
$$\mathcal{T}(L(x \to \Theta)) = \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi$$

Transport of Light



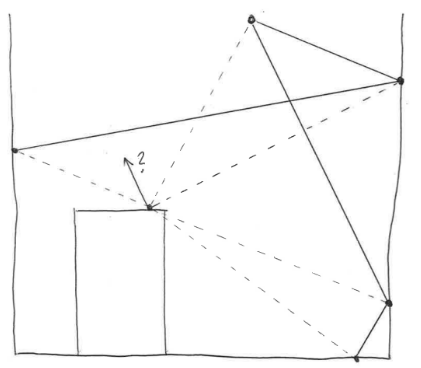
$$L(x \to \Theta) = L_e(x \to \Theta) + \ldots$$

Transport of Light



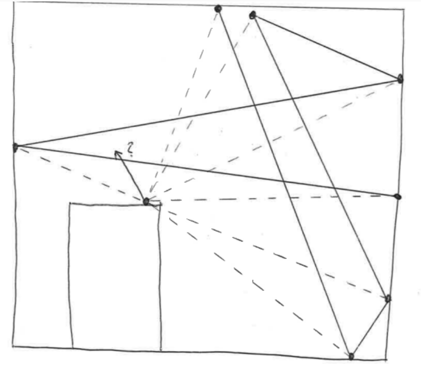
$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \ldots$$

Transport of Light



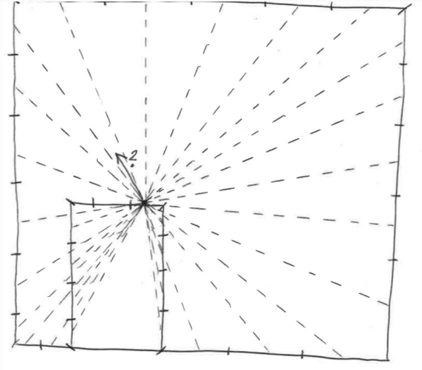
$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \langle T, TL_e \rangle + \ldots$$

Transport of Light



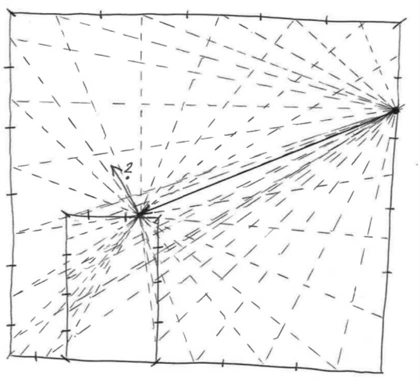
$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \langle T, TL_e \rangle + \langle T, TTL_e \rangle + \ldots$$

Transport of Light



$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \langle T, TL_e \rangle + \langle T, TTL_e \rangle + \dots$$

Transport of Light



$$L(x \to \Theta) = L_e(x \to \Theta) + \langle T, L_e \rangle + \langle T, TL_e \rangle + \langle T, TTL_e \rangle + \dots$$

That Integral

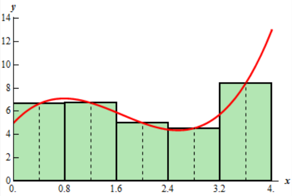
$$\mathcal{T}(L(x \to \Theta)) = \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi$$

That Integral

Nature laughs at the difficulties of integration![1]

[1] Kajiya, James T., "The rendering equation", Siggraph 1986

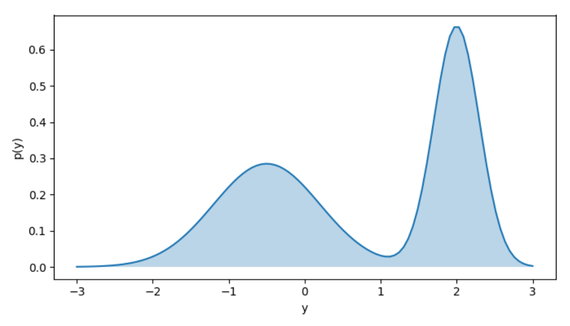
Approximate Integration (Stochastic)



$$\int f(x)\mathrm{d}x \approx \sum_{i=2}^{N} \frac{1}{2}(f(x_i))(x_i - x_{i-1})$$

- Approximate the integral with a sum, in the limit we will be exact
- A whole research field called *sampling*
- Often stochastic evaluation of integrand
- *Remeber the the curse of dimensionality*

Random Variable



- Random variable, is a stochastic variable that follows a distribution
- Random does not mean max entropy

Expected Value

Expected Value

$$\mathbb{E}[x] = \int x p(x) \mathrm{d}x$$

Ex: Fair dice

$$\mathbf{X} = \{1, 2, 3, 4, 5, 6\}$$
$$p(x_i) = \frac{1}{6}$$
$$\mathbb{E}[x] = \sum_{x_i \in \mathbf{X}} x_i p(x_i) = \frac{1}{6} = \frac{1}{6}(1 + 2 + 3 + 4 + 5 + 6) = 3.5$$

Expected Value

- The expected value is characteristic of a distribution
- It is a location parameters
- Also referred to as *the mean*
- Idea: *can we compute the expected value of the integral?*

Expected Value

Variance

$$\sigma^2(x) = \mathbb{E}\left[(x - \mathbb{E}[x])^2\right]$$

Ex: Fair dice

$$\begin{aligned}
\mathbf{X} &= \{1, 2, 3, 4, 5, 6\} \\
p(x_i) &= \frac{1}{6} \\
\sigma^2(x) = \mathbb{E}\left[(x - \mathbb{E}[x])^2\right] &= \sum_{x_i \in \mathbf{X}} (x_i - 3.5)^2 p(x_i) \\
&= \frac{1}{6}((1 - 3.5)^2 + (2 - 3.5)^2 + \ldots) = 2.91
\end{aligned}$$

Idea

- The variance is characteristic of a distribution
- It says the "spread" of the distribution around its mean
  - we use this when we reason all the time as a measure of uncertainty
- Idea: *can we use the variance of an integration to see how certain we are about its value?*

Monte-Carlo Estimator

$$I = \int f(x)\mathrm{d}x$$

$$\langle I \rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

Monte-Carlo Estimator

$$I = \int f(x)\mathrm{d}x$$

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

$$\mathbb{E}[\langle I \rangle] = \mathbb{E}\left[\frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}\right]$$

Monte-Carlo Estimator

$$I = \int f(x)\mathrm{d}x$$

$$\langle I \rangle = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}$$

$$\mathbb{E}[\langle I \rangle] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)}\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\left[\frac{f(x_i)}{p(x_i)}\right]$$

Monte-Carlo Estimator

$$I = \int f(x)\mathrm{d}x$$

$$\langle I \rangle = \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}$$

$$\begin{aligned}\mathbb{E}[\langle I \rangle] &= \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\frac{f(x_i)}{p(x_i)}\right] \\ &= \frac{N}{N}\int \frac{f(x)}{p(x)}p(x)\mathrm{d}x = \int f(x)\mathrm{d}x = I\end{aligned}$$

Monte-Carlo Estimator

$$
\begin{aligned}
I &= \int f(x)\mathrm{d}x \\
\langle I \rangle &= \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)} \\
\mathbb{E}[\langle I \rangle] &= \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right] = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}\left[\frac{f(x_i)}{p(x_i)}\right] \\
&= \frac{N}{N}\int \frac{f(x)}{p(x)}p(x)\mathrm{d}x = \int f(x)\mathrm{d}x = I
\end{aligned}
$$

- The expected value of the approximation is the value of the integral

Monte-Carlo Estimator

$$\sigma^2(\langle I \rangle) = \sigma^2 \left(\frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \right)$$

Monte-Carlo Estimator

$$\sigma^2(\langle I \rangle) = \sigma^2 \left(\frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \right)$$

$$\left\{ \sigma^2(a \cdot x) = a^2 \sigma^2(x) \right\}$$

Monte-Carlo Estimator

$$\sigma^2(\langle I \rangle) = \sigma^2 \left(\frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \right) = \frac{1}{N^2} \sigma^2 \left(\sum_{i=1}^{N} \frac{f(x_i)}{p(x_i)} \right)$$

Monte-Carlo Estimator

$$\sigma^2(\langle I \rangle) = \sigma^2\left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right) = \frac{1}{N^2}\sigma^2\left(\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right)$$

$$\left\{\sigma^2(\sum_i^N x_i) = \sum_i^N \sigma^2(x_i)\right\}$$

Monte-Carlo Estimator

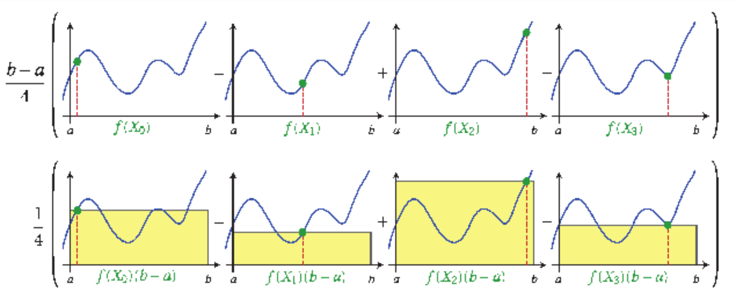
$$\sigma^2(\langle I \rangle) = \sigma^2\left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right) = \frac{1}{N^2}\sigma^2\left(\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right)$$
$$= \frac{1}{N^2}\left(\sum_{i=1}^{N}\sigma^2\ \frac{f(x_i)}{p(x_i)}\right)$$

Monte-Carlo Estimator

$$
\begin{aligned}
\sigma^2(\langle I \rangle) = \sigma^2\left(\frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right) &= \frac{1}{N^2}\sigma^2\left(\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\right) \\
&= \frac{1}{N^2}\left(\sum_{i=1}^{N}\sigma^2\ \frac{f(x_i)}{p(x_i)}\right) \\
\sigma(\langle I \rangle) &= \frac{1}{\sqrt{N}}\sigma\left(\sum_{i}\frac{f(x_i)}{p(x_i)}\right)
\end{aligned}
$$

- The standard deviation of the estimator will converge as $\mathcal{O}(\sqrt{N})$

Monte-Carlo Integration[2]



[2] Thesis Appendix of Wojciech Jarosz

Monte-Carlo Integration Summary

Summary

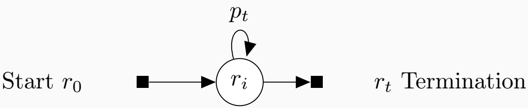
- We treat the sample location as a random variable
- This means that the estimator is also a random varialbe
- If we draw sufficiently many samples the estimator will converge
- It is not terribly quick and converges with $\mathcal{O}(\sqrt{N})$

Monte Carlo Path Tracing

Path Tracing

- Path Tracing is the recursive way to solve the rendering equation
- Just keep shooting rays and the image will look better
- Will eventually get you "everything" but it will take time

Markov Chain



Step 1 Create a random particle in state i with probabilit p_i^0

Step 2 With probability $p_i^* = 1 - \sum_{j=0} p_{ij}$ terminate in state i

- if terminate go to Step 1

Step 3 Randomly select new state j according to transition probability p_{ij} and go to Step 2

Monte Carlo Path Tracer

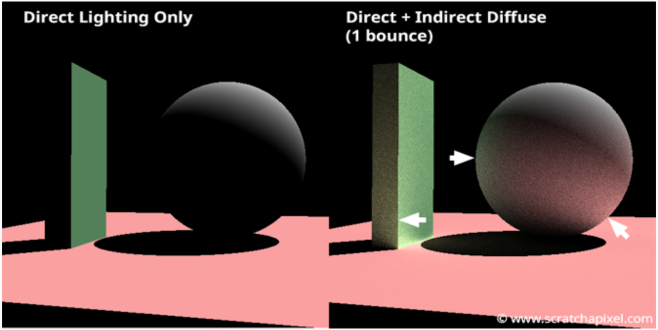
1. Choose a ray that goes through the pixel
   - $\lambda = 1.0$
2. Find intersection
   - $\lambda = \rho(\cdot)$ (reflectance function)
   - choose if returning emitted light or calculating reflected

Emitted `return` $\lambda \cdot \frac{1}{p} L_e$

Reflected `return` $\lambda \cdot \frac{1}{p}$TraceRay

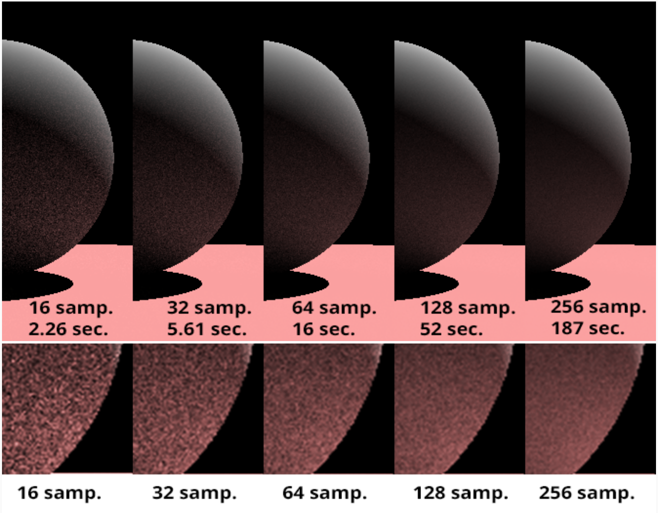
3. Pixel value is average of each ray
4. Add Markov Chain to decide if to traverse further

Summary

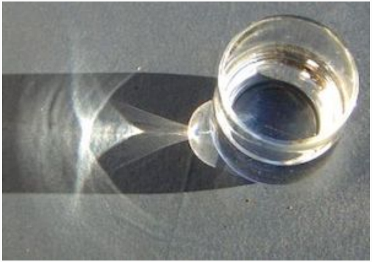


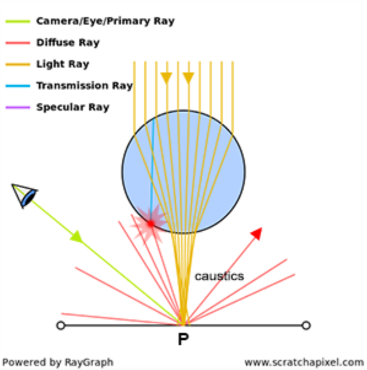
Direct Lighting Only
Direct + Indirect Diffuse
(1 bounce)
© www.scratchapixel.com

Summary

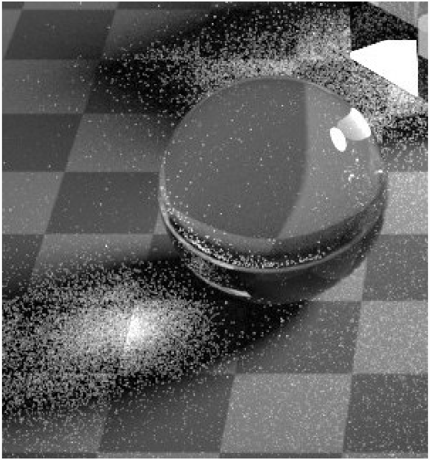

16 samp.
2.26 sec.
32 samp.
5.61 sec.
64 samp.
16 sec.
128 samp.
52 sec.
256 samp.
187 sec.
16 samp.
32 samp.
64 samp.
128 samp.
256 samp.

Summary




Camera/Eye/Primary Ray
Diffuse Ray
Light Ray
Transmission Ray
Specular Ray
caustics
P
Powered by RayGraph
www.scratchapixel.com

Summary



Photon Mapping

Photon Mapping

- Developed by Henrik Jensen TU Copenhagen
- First proposed 1993
- Propsed as a means to speed up path tracing
- Path tracing often results in high-frequent noise which is very apparent while photon mapping results in low-frequent noise

Photon Mapping

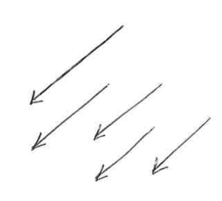
Pass 1 photon shooting stage

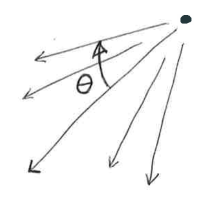
- From each lightsource distribute photons around the scene
- Forward raytracer

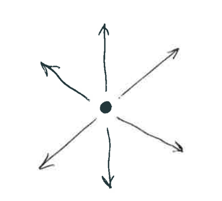
Pass 2 gathering stage

- shoot camera rays and gather the light that has been distributed
- Backward raytracer

Photon Mapping







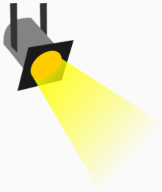
$$\Phi_{\text{photon}} = \frac{\Phi_{\text{light}}}{N}$$

Projection Maps

$$\Phi_{\text{photon}} = \frac{\Phi_{\text{light}}}{N} \cdot \frac{\text{cells with objects}}{\text{total number of cells}}$$

- Rather than shooting photons blindly we can be a bit clever about it
- Create a map which contains the world as seen from the lightsource
- Discritisise the map into cells
- Use this as a mask to avoid performing intersection checks

Shooting Photons



Shooting photons is exactly the same as shooting rays, however, we now shoot flux while in the path tracer we gathered radiance

Photon Mapping

- When a photon hits an object it can either be,
    1. Reflected
    2. Refracted
    3. Absorbed
- To determine what happens we play . . .

Russian Roulette

Russian Roulette

- Associate each material with a reflection coefficient
  - *s* - specular reflection
  - *d* - diffuse reflection

Russian Roulette

- Associate each material with a reflection coefficient
  - s - specular reflection
  - d - diffuse reflection
- Draw a random number $\eta \in [0, 1]$
  - $\eta \in [0, d] \rightarrow$ diffuse reflection
  - $\eta \in [d, d + s] \rightarrow$ specular reflection
  - $\eta \in [d + s, 1] \rightarrow$ absorption

RGB

$$P_r = \max(d_r + s_r, d_g + s_g, d_b + s_b)$$
$$P_a = 1 - P_r$$

- Diffuse reflection

$$P_d = \frac{d_r + d_g + d_b}{d_r + d_g + d_b + s_r + s_g + s_b} P_r$$

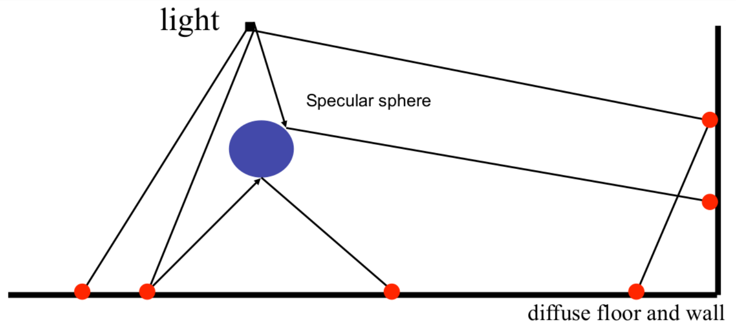
- Specular reflection

$$P_s = \frac{s_r + s_g + s_b}{d_r + d_g + d_b + s_r + s_g + s_b} P_r$$

RGB

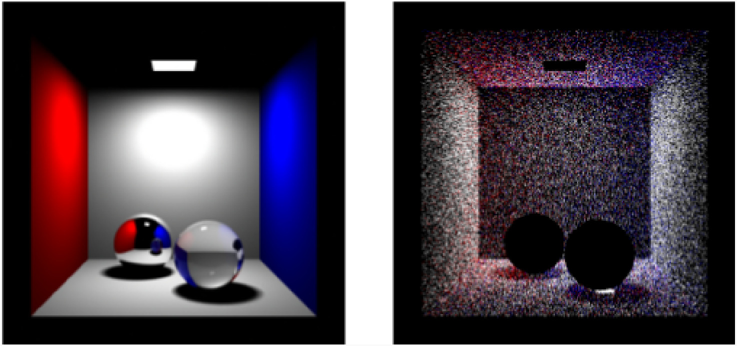
$$\begin{aligned} \eta \in [0, P_d] &\rightarrow \text{diffuse reflection} \\ \eta \in [P_d, P_s + P_d] &\rightarrow \text{specular reflection} \\ \eta \in [P_s + P_d, 1] &\rightarrow \text{absorption} \end{aligned}$$

Photon Maps



- We keep tracing the photon until it gets absorbed
- This map is called the photon map
- Only store at diffuse locations

Photon Maps[1]



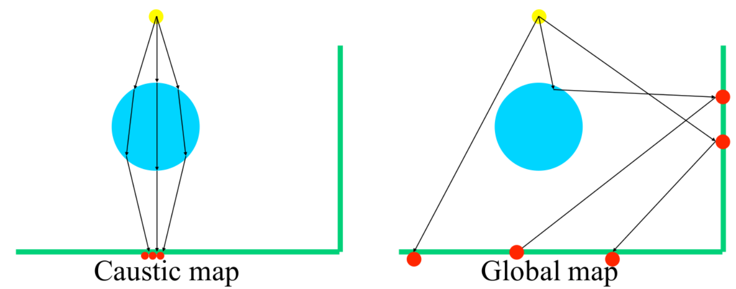
Photon Storage

Caustic map Create a projection map that specifically "targets" areas which could generate caustics

Volume map Create a projection map for participating media where interactions happens randomly

Global map A projection map that contains any geometry

Photon Maps



Pass II

Factorisation

BRDF

$$f_r(x, \Psi \to \Theta) = f_{r,s}(x, \Psi \to \Theta) + f_{r,d}(x, \Psi \to \Theta)$$

- factorise BRDF in specular and diffuse components

Light

$$L(x \leftarrow \Psi) = L_l(x \leftarrow \Psi) + L_c(x \leftarrow \Psi) + L_d(x \leftarrow \Psi)$$

- factorise *incoming* radiance in direct, caustic and indirect

Rendering Equation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_{\Psi} =$$

Rendering Equation

$$L(x \to \Theta) = L_e(x \to \Theta) + \int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi =$$

$$= L_e(x \to \Theta) + \int_{\Omega_x} (f_{r,s} + f_{r,d}) L_l cos(\cdot) d\omega_\Psi + \int_{\Omega_x} f_{r,s}(L_c + L_d) cos(\cdot) d\omega_\Psi +$$

$$\int_{\Omega_x} f_{r,d} L_c cos(\cdot) d\omega_\Psi + \int_{\Omega_x} f_{r,d} L_d cos(\cdot) d\omega_\Psi$$

Rendering

$$\int_{\Omega_x} (f_{r,s} + f_{r,d}) L_l cos(\cdot) d\omega_\Psi$$

- Direct illumination, just as in bog-standard raytracer

$$\int_{\Omega_x} f_{r,s} (L_c + L_d) cos(\cdot) d\omega_\Psi$$

- Specular/glossy reflection very "peaked" so use normal path tracing

Rendering

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \rightarrow \Theta)}$$

Rendering

$$L(x \to \Theta) = L_e(x \to \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \to \Theta)}$$

$$L(x \leftarrow \Psi) = \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA}$$

Rendering

$$L(x \to \Theta) = L_e(x \to \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \to \Theta)}$$

$$L_r(x \to \Theta) = \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA} cos(\mathbf{n}_x, \Psi) d\Psi$$

Rendering

$$L(x \to \Theta) = L_e(x \to \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \to \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_\Psi}_{L_r(x \to \Theta)}$$

$$\begin{aligned} L_r(x \to \Theta) &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA} cos(\mathbf{n}_x, \Psi) d\Psi \\ &= \int_{\Omega_x} f_r(x, \Psi \to \Theta) \frac{d^2\Phi(x, \Psi)}{dA} \end{aligned}$$

Rendering

$$L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \underbrace{\int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) L(x \leftarrow \Psi) cos(\mathbf{n}_x, \Psi) d\omega_{\Psi}}_{L_r(x \rightarrow \Theta)}$$

$$\begin{aligned} L_r(x \rightarrow \Theta) &= \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) \frac{d^2\Phi(x, \Psi)}{cos(\mathbf{n}_x, \Psi) d\Psi dA} cos(\mathbf{n}_x, \Psi) d\Psi \\ &= \int_{\Omega_x} f_r(x, \Psi \rightarrow \Theta) \frac{d^2\Phi(x, \Psi)}{dA} \\ &\approx \sum_{p=1}^{N} f_r(x, \Psi \rightarrow \Theta) \frac{\Delta\Phi(x, \Psi)}{\Delta A} \end{aligned}$$

$$L(x \to \Theta) \approx L_e(x \to \Theta) + \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \frac{\Delta\Phi_p(x, \Psi)}{\Delta A}$$

- If we assume that each photon have flux $\Delta\Phi_p$ we can "approximate" the sum above by finding the N closest photons to x and summing their flux
- *Think of this as expanding a sphere around x*

Rendering

$$L(x \to \Theta) \approx L_e(x \to \Theta) + \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \frac{\Delta\Phi_p(x, \Psi)}{\Delta A}$$

$$= L_e(x \to \Theta) + \frac{1}{\pi r^2} \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \Delta\Phi_p(x, \Psi)$$

- If we assume that each photon have flux $\Delta\Phi_p$ we can "approximate" the sum above by finding the N closest photons to x and summing their flux
- *Think of this as expanding a sphere around x*

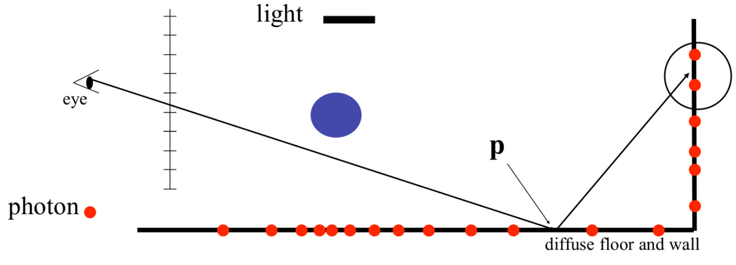
Rendering

$$L(x \to \Theta) \approx L_e(x \to \Theta) + \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \frac{\Delta\Phi_p(x, \Psi)}{\Delta A}$$

$$= L_e(x \to \Theta) + \frac{1}{\pi r^2} \sum_{p=1}^{N} f_r(x, \Psi \to \Theta) \Delta\Phi_p(x, \Psi)$$

- If we assume that each photon have flux $\Delta\Phi_p$ we can "approximate" the sum above by finding the N closest photons to x and summing their flux
- *Think of this as expanding a sphere around x*
- This will be very wrong if the area of accumulation is not flat

Rendering



Rendering

$$\int_{\Omega_x} f_{r,d} L_c cos(\cdot) d\omega_\Psi = \frac{1}{\pi r^2} \sum_{p=1}^{N} f_{r,d}(x, \Psi \to \Theta) \Delta\Phi_p^c(x, \Psi)$$

- Caustics will be evaluated with the Caustic Photon Map

$$\int_{\Omega_x} f_{r,d} L_d cos(\cdot) d\omega_\Psi = \frac{1}{\pi r^2} \sum_{p=1}^{N} f_{r,d}(x, \Psi \to \Theta) \Delta\Phi_p^g(x, \Psi)$$

- Diffuse light will be evaluated with the Global Photon Map

Optimisations

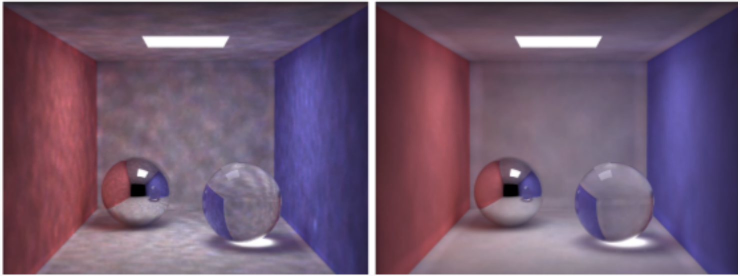
Datastructures

- When we render the scene we compute a lot of distances
- Store the photon-maps in a tree to save look-up time
- This is where the big speed-up can be done and is worth looking into

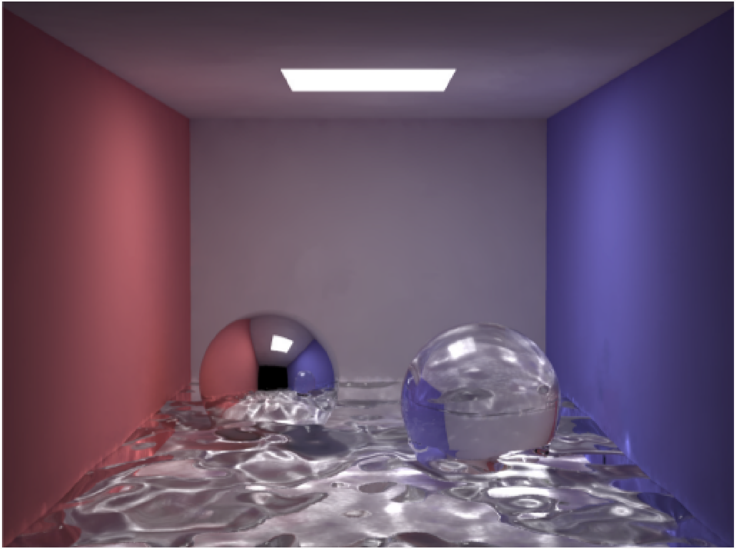
Irradiance Caching

- Indirect light is more likely to be "smooth"
- Cache computations for certain points in the scene
- Interpolate out the missing values

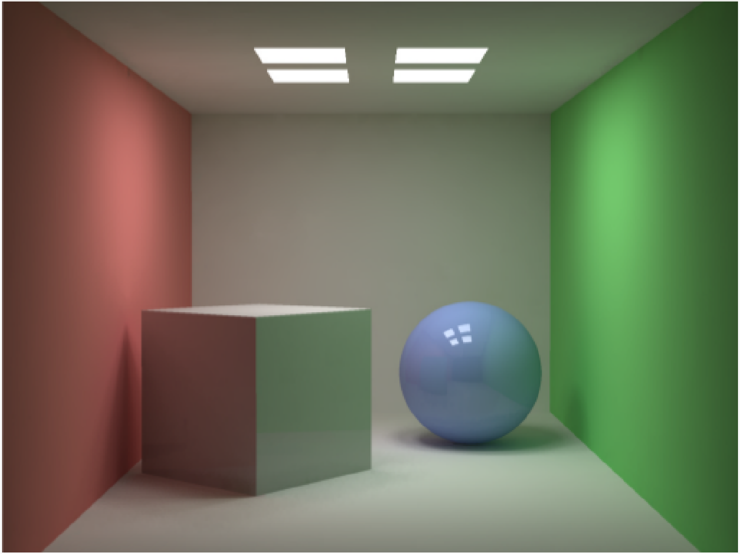
Results[1]



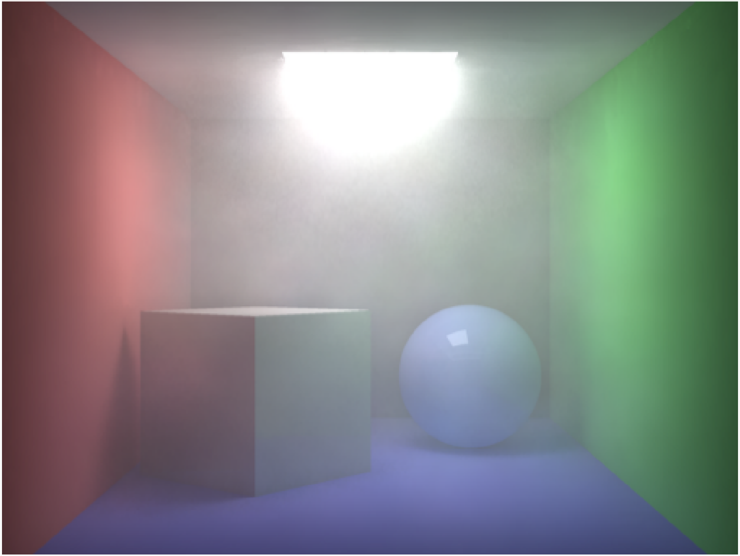
Results[1]



Results[1]



Results[1]



Summary

Summary

- Two-pass rendering strategy
- Allows to capture things like caustics
- More "hacky" compared to path tracing or radiosity
- Uses several different approaches together

Submission

- Deadline 27th of April
- Report
- Viva

Summary

Global Illumination

Radiosity Completely diffuse surfaces

- Does colour bleeding really well
- Equation system
- View point independent but requires discretisation

Path Tracing Keep tracing light paths in reverse directing

- Correct but slow
- High frequent noise

Photon Mapping Two pass, forward and backward tracing

- Capable of doing caustics easy
- Low frequent noise
- "less" correct

Next Time

Lecture Monday 27th of April

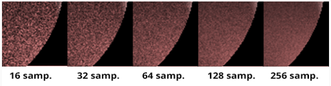
- Unit Summary
- Coursework wrap-up/VIVA etc.
- What to do next
- Thesis work in Computer Graphics
- Work in Computer Graphics

END

eof

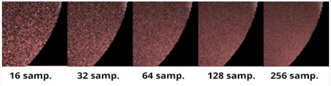
Appendix

Monte-Carlo Integration



$$I = \int f(x)\mathrm{d}x$$

Monte-Carlo Integration



$$
\begin{aligned}
I &= \int f(x)\mathrm{d}x \\
&\approx \int \mathrm{d}x \frac{1}{|\mathbf{X}|} \sum_{x_i \in \mathbf{X}} f(x_i)
\end{aligned}
$$