Lecture: Global Illumination

Kavita Bala
Associate Professor, Cornell University
On Sabbatical at MIT
Light Paths
Full Global Illumination

… find paths between sources and surfaces to be shaded
Classic Whitted Ray Tracing

- Shoot ray from eye
  - Find closest visible object
- For each visible point
  - shoot one shadow ray
  - shoot one reflected/refracted ray
Classic Whitted Ray Tracing

- Point lights
  - Unrealistic
  - Hard shadows

- BRDF is simple
  - Pure specular

- Ignores many paths
  - Including diffuse inter-reflections
  - Does not solve the Rendering Equation
How do we get all paths?
What problem should we solve?

- **Goal:**
  - to compute steady-state radiance values in scene

- **Rendering equation:**
  - mathematical formulation of problem that global illumination algorithms must solve
Rendering Equation (RE)

- RE describes energy transport in scene

- Input
  - Light sources
  - Surface geometry
  - Reflectance characteristics of surfaces

- Output: value of radiance at all surface points in all directions
Rendering Equation

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + L_r(x \rightarrow \Theta) \]
Rendering Equation

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + L_r \]
Rendering Equation

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int L(x \leftarrow \Psi) \text{ hemisphere} \]
\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\text{hemisphere}} L(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(N_x, \Psi) d\omega_\Psi \]
Rendering Equation

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int L(x \leftarrow \Psi) f_r(x, \Psi \leftrightarrow \Theta) \cos(N_x, \Psi) d\omega_\Psi \]
Evaluating the Rendering Eqn

- Radiance is hard to evaluate

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi \]

- Sample many paths: integrate over all incoming directions
Why Is it Hard to Evaluate?

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi \]

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} L_e(y \rightarrow -\Psi) f_r(\Psi \leftrightarrow \Theta) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi + \]

\[ \int_{\Omega_x} \int_{\Omega_y} f_r(\Psi' \rightarrow -\Psi) \cos(\Psi', n_y') L(y \leftarrow \Psi') d\omega_\Psi' f_r(\Psi \leftrightarrow \Theta) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi \]
Why Monte Carlo?

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \]

\[ \int_{\Omega_x} L_e(y \rightarrow -\Psi) f_r(\Psi \leftrightarrow \Theta) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi + \]

......

• Analytical integration is difficult

• Therefore, need numerical techniques
Monte Carlo Integration

- Numerical tool to evaluate integrals
- Use sampling
- Stochastic errors
- Unbiased
  - on average, we get the right answer!
Numerical Integration

• A one-dimensional integral:

\[ I = \int_{a}^{b} f(x) \, dx \]
Deterministic Integration

- Quadrature rules:

\[ I = \int_{a}^{b} f(x) \, dx \]

\[ \approx \sum_{i=1}^{N} w_i f(x_i) = \sum_{i=1}^{N} \frac{1}{N} f(x_i) \]
Similar idea, but random samples:
Generate $N$ random samples $x_i$

Estimator:

$$\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} f(\bar{x}_i)$$
Monte Carlo Integration

See why it works for one sample:

\[ I = \int_{a}^{b} f(x) \, dx \]

\[ I_{\text{prim}} = f(\bar{x}) \]

\[ E(I_{\text{prim}}) = \int_{0}^{1} f(x) \, p(x) \, dx = \int_{0}^{1} f(x) \, 1 \, dx = I \]

Unbiased estimator!
Monte Carlo Integration

• Expected value of estimator

\[
E[\langle I \rangle] = E\left[\frac{1}{N} \sum_{i}^{N} f(x_i)\right] = \frac{1}{N} \int (\sum_{i}^{N} f(x_i)) p(x) \, dx
\]

\[
= \frac{1}{N} \sum_{i}^{N} \int f(x) \, dx
\]

\[
= \frac{N}{N} \int f(x) \, dx = I
\]

– on ‘average’ get right result: unbiased

• Standard deviation \( \sigma \) is a measure of the stochastic error

\[
\sigma^2 = \frac{1}{N} \int_{a}^{b} \left[ \frac{f(x)}{p(x)} - I \right]^2 \ p(x) \, dx
\]
MC Integration - Example

- Integral

\[ I = \int_{0}^{1} 5x^4 \, dx = 1 \]

- Uniform sampling

- Samples :

\[
\begin{align*}
    x_1 &= .86 & \langle I \rangle &= 2.74 \\
    x_2 &= .41 & \langle I \rangle &= 1.44 \\
    x_3 &= .02 & \langle I \rangle &= 0.96 \\
    x_4 &= .38 & \langle I \rangle &= 0.75
\end{align*}
\]
MC Integration - Example

- Integral

\[ I = \int_{0}^{1} 5x^4 \, dx = 1 \]

- Stochastic error

- Variance
  - What is it?
MC Integration: 2D

\[ I_{\text{sec}} = \frac{1}{N} \sum_{i=1}^{N} \frac{f(\bar{x}_i, \bar{y}_i)}{p(\bar{x}_i, \bar{y}_i)} \]
Monte Carlo Integration - 2D

- MC Integration works well for higher dimensions
- Unlike quadrature

\[
I = \int_{a}^{b} \int_{c}^{d} f(x,y) \, dx \, dy
\]

\[
\langle I \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f(x_i, y_i)}{p(x_i, y_i)}
\]
MC Advantages

• Convergence rate of $O\left(\frac{1}{\sqrt{N}}\right)$
• Simple
  – Sampling
  – Point evaluation
  – Can use black boxes
• General
  – Works for high dimensions
  – Deals with discontinuities, crazy functions,…
Importance Sampling

• Why not just uniformly?

• Better use of samples by taking more samples in ‘important’ regions, i.e. where the function is large
Numerical Example
\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi \]
Radiance Evaluation

• Many different light paths contribute to single radiance value
  – many paths are unimportant

• Tools we need:
  – generate the light paths
  – sum all contributions of all light paths
  – clever techniques to select important paths
Assumptions: black boxes

- Can query the scene geometry and materials
  - surface points
  - light sources
  - visibility checks
  - tracing rays

\[ N = ? \]
\[ f_r = ? \]
\[ f_r(x, \Theta \leftrightarrow \Psi) = ? \]
\[ V(x, z) = 0 \text{ or } 1 \]
\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi \]

Value we want

\[ = L_e + \int_{\Omega_x} \cdot f_r \cdot \cos \]

function to integrate over all incoming directions over the hemisphere around x
How to compute?

$L(x \rightarrow \Theta) = \square$

Check for $L_e(x \rightarrow \Theta)$

Now add $L_r(x \rightarrow \Theta) =$

$$\int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_{\Psi}$$

$L = \square$
How to compute?

Monte Carlo!
Generate random directions $\Psi_i$

$$\langle L \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(...) \cdot L(x \leftarrow \Psi_i) \cdot \cos(...)}{p(\Psi_i)}$$

- evaluate brdf
- evaluate cosine term
- evaluate $L(x \leftarrow \Psi_i)$
How to compute?

- evaluate $L(x \leftarrow \Psi_i)$?

- Radiance is invariant along straight paths

- $vp(x, \Psi_i) =$ first visible point

- $L(x \leftarrow \Psi_i) = L(vp(x, \Psi_i) \rightarrow \Psi_i)$
How to compute? Recursion ... 

- Recursion ....
- Each additional bounce adds one more level of indirect light
- Handles ALL light transport
- “Stochastic Ray Tracing”
When to end recursion?

- Contributions of further light bounces become less significant
- If we just ignore them, estimators will be biased!
  - Use probabilistic Russian Roulette to terminate recursion

Kavita Bala, Cornell University
Algorithm so far ...

- Shoot viewing ray through each pixel
- Shoot # indirect rays, sampled over hemisphere
- Terminate recursion using Russian Roulette
Algorithm

\[ L^e = 1.234 \]

\[ L' = ? \]
Stochastic Ray Tracing

• Parameters?
  – # starting rays per pixel
  – # random rays for each surface point (branching factor)

• Path Tracing
  – Branching factor == 1
Pixel Anti-Aliasing

- Compute radiance only at center of pixel: jaggies

- Simple box filter:

\[ L = \int_{Pixel} L(x) \, dx \]

- … evaluate using MC
Path tracing

1 ray / pixel  10 rays / pixel  100 rays / pixel

- Pixel sampling + light source sampling folded into one method

Kavita Bala, Cornell University
Comparison

1 centered viewing ray
100 random shadow rays per viewing ray

100 random viewing rays
1 random shadow ray per viewing ray
Sample Other Domains

\[ \text{Pixel} = \int_{\text{pixel area}} \int_{\text{aperture}} \int_{\text{time}} \int_{\text{volume}} L(...) \]
Kitchen

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polygons</td>
<td>388,552</td>
</tr>
<tr>
<td>Light Points</td>
<td>55,189</td>
</tr>
<tr>
<td>Gather Points</td>
<td>100</td>
</tr>
<tr>
<td>Gather/Light Pairs</td>
<td>5,518,900</td>
</tr>
<tr>
<td>Cut Size</td>
<td>936 (0.02%)</td>
</tr>
</tbody>
</table>
Roulette

Polygons: 151,752
Light Points: 23,000
Gather Points: 306
Gather/Light Pairs: 7,047,430
Cut Size: 174 (0.002%)
<table>
<thead>
<tr>
<th>Tableau</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polygons: 630,843</td>
</tr>
<tr>
<td>Light Points: 13,000</td>
</tr>
<tr>
<td>Gather Points: 180</td>
</tr>
<tr>
<td>Gather/Light Pairs: 234,000</td>
</tr>
<tr>
<td>Cut Size: 447 (0.2%)</td>
</tr>
</tbody>
</table>

Kavita Bala, Cornell University
Algorithm so far ...

- Shoot # viewing rays through each pixel
  - Distributed over pixel, time, camera aperture

- Shoot # indirect rays, sampled over hemisphere
  - Path tracing shoots only 1 indirect ray

- Terminate recursion using Russian Roulette
Performance/Error

• Want better quality with smaller number of samples
  – Fewer samples/better performance
  – Stratified sampling
  – Quasi Monte Carlo: well-distributed samples

• Faster convergence
  – Importance sampling
Path Tracing

Sample hemisphere

- Importance Sampling: compute direct illumination separately!
Direct Illumination

• Paths of length 1 only, between receiver and light source
With Indirect Illumination
Direct Illumination Sampling

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_\Psi \]

Radiance from light sources + radiance from other surfaces

\[ = L_e + \int_{\Omega_x} f_r \cdot \cos \]
Next Event Estimation

\[ L(x \rightarrow \Theta) = L_e + L_{direct} + L_{indirect} \]

\[ = L_e + \int_{\Omega_x} \cdot f_r \cdot \cos + \int_{\Omega_x} \cdot f_r \cdot \cos \]

• So ... sample direct and indirect with separate MC integration
Rendering Equation: all surfaces

\[ L(x \rightarrow \Theta) = L_e (...) + \int_{\mathcal{A}} f_r (...) \cdot L(y \rightarrow -\Psi) \cdot \frac{\cos\theta_x \cdot \cos\theta_y}{r_{xy}^2} \cdot V(x, y) \, dA_y \]
Algorithm
Algorithm
Algorithm

→ a variant of path tracing
Comparison

Without N.E.E.  With N.E.E.

16 samples/pixel
Rays per pixel

1 sample/pixel

4 samples/pixel

16 samples/pixel

256 samples/pixel
Direct Illumination

\[
L(x \rightarrow \Theta) = \int_{A_{\text{source}}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L(y \rightarrow \Psi) \cdot G(x, y) \cdot dA_y
\]

\[
G(x, y) = \frac{\cos(n_x, \Theta) \cos(n_y, \Psi) \text{Vis}(x, y)}{r_{xy}^2}
\]

hemisphere integration

area integration
Generating direct paths

- Pick surface points $y_i$ on light source
- Evaluate direct illumination integral

\[
\langle L(x \to \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r(...)L(...)G(x, y_i)}{p(y_i)}
\]
More points ... 

\[ p(y) = \frac{1}{\text{Area}_{\text{source}}} \]
Even more points ...

36 shadow rays

100 shadow rays
Strategies for picking light

- **Uniform**\[ p_L(k) = \frac{1}{M} \]

- **Area**\[ p_L(k) = \frac{A_k}{\sum A_k} \]

- **Power**\[ p_L(k) = \frac{P_k}{\sum P_k} \]

- More sophisticated based on visibility
Importance Sampling: BRDF
Importance Sampling: BRDF

5 Samples/Pixel

MIT EECS 6.837

Slide courtesy of Jason Lawrence
Importance Sampling: BRDF

25 Samples/Pixel

MIT EECS 6.837

Slide courtesy of Jason Lawrence
BRDF Sampling

Uniform: 5 samples               25 samples                   75 samples
BRDF: 5 samples                  25 samples                   75 samples
Comparison

With importance sampling (brdf on sphere)  
Without importance sampling

Kavita Bala, Cornell University
What if Regular Samples
Classic ray tracing?

• Shoot shadow-rays (direct illumination)

• Shoot perfect specular rays only for indirect

• Ignores many paths
  – Does not solve the rendering equation
Pure Path Tracing

• Advantages
  – No need for meshing
  – General surfaces – requires ray intersections
  – *Unbiased* estimates

• Disadvantages
  – Noisy! Every point is independent
  – Starts from scratch – does not exploit coherence
Path Re-Use

• What is coherence?
  – Nearby values are similar to what we want
  – Indirect illumination is smooth
Unbiased vs. Consistent

• Unbiased
  – No systematic error
  – $E[I_{\text{estimator}}] = I$
    ▪ Better results with larger $N$

• Consistent
  – Converges to correct result with more samples
  – $E[I_{\text{estimator}}] = I + \varepsilon$ where $\lim_{N \to \infty} \varepsilon = 0$
Biased Methods

• Store information (caching)
  – Better type of noise: blurring

• Greg Ward’s Radiance
• Photon Mapping
• Instant Radiosity
• Lightcuts/Multidimensional lightcuts
Irradiance Caching

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

• “Cache” smooth irradiance when possible
  – Path reuse (assumes diffuse surfaces)
• When we need irradiance at a new point
  – Interpolate using cached samples
• Basic path tracing algorithm, but
  – If hit diffuse surface
    ▪ Always compute direct lighting explicitly
    ▪ Build an estimate of irradiance at that point
      • If estimate is good, use it
      • If not, use path tracing to estimate the irradiance and store it in octree
Photon Map

- Build on irradiance caching
- Use bidirectional ray tracing

Caustic: LS+ D E
Photon Map

• 2 passes:
  – shoot “photons” (light-rays) and record any hit-points
  – shoot viewing rays, collect information from stored photons
Pass 1: shoot photons
Indirect Visualization

Figure 3: The Museum scene

Figure 4: Direct visualization of the global photon map in the Museum scene
Photon Map Results
Scalable Rendering

- Lightcuts/Multidimensional Lightcuts
  - Scale to number of lights
  - Effects like motion blur, depth of field
Summary

• Monte Carlo Rendering
  – The definitive solution for Global Illumination
  – But slow

• Fast techniques to produce high quality images: active area of research
Resources