Lecture 8: Monte Carlo
Rendering

CS 6620, Spring 2009
Kavita Bala
Computer Science
Cornell University

MC applied to RE

\[ L(x \rightarrow \Theta) = L_c(x \rightarrow \Theta) + \int_{\Omega_x} f_r(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos(\Psi, n_x) \cdot d\omega_y \]
How to compute?

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

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

Now add $L_f(x \rightarrow \Theta) = \int_{\Omega} f_i(\Psi \leftrightarrow \Theta) \cdot L(x \leftrightarrow \Psi) \cdot \cos(\Psi, n_i) \cdot d\omega_i$

Generate random directions $\Psi_i$

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

How to compute? Recursion ...

- Recursion ….
- Each additional bounce adds one more level of indirect light
- Handles ALL light transport
- “Stochastic Ray Tracing”
Russian Roulette

Integral

\[ I = \int_0^1 f(x)dx = \frac{1}{P} \int_0^p f(x)Pdx = \frac{1}{P} \int_0^1 \frac{f(y/P)}{P} dy \]

Estimator

\[ \langle I_{\text{roulette}} \rangle = \begin{cases} \frac{f(x_i)}{P} & \text{if } x_i \leq P, \\ 0 & \text{if } x_i > P. \end{cases} \]

Variance \( \sigma_{\text{roulette}} > \sigma \)

Pixel Anti-Aliasing

- Compute radiance only at center of pixel: jaggies
- Simple box filter:
  - ... evaluate using MC
Stochastic Ray Tracing

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

• Path Tracing
  – Branching factor == 1

Path tracing

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

• Pixel sampling + light source sampling folded into one method
Comparison

1 centered viewing ray
100 random shadow rays per viewing ray

100 random viewing rays
1 random shadow ray per viewing ray

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: next-event estimation
Stratified and Importance?

Higher Dimensions

- Stratified grid sampling:
  - $N^d$ samples

- N-rooks sampling:
  - $N$ samples
Quasi Monte Carlo

- Eliminates randomness to find well-distributed samples
  - Why? Avoid clumping
  - Why? Has better convergence properties

Quasi-Monte Carlo (QMC)

- Notions of variance, expected value don’t apply: why?

- Introduce the notion of discrepancy
  - Discrepancy mimics variance
  - Need a low discrepancy sequence
  - E.g., subset of unit interval [0,x]
    - Of N samples, n are in subset
    - Discrepancy: |x-n/N|
  - Mainly: “it looks random”
Example: Halton

- Radical inverse $\phi_p(i)$ for primes $p$
- Reflect digits (base $p$) about decimal point
  \[ \phi_2(i): 111010_2 \rightarrow 0.010111 \]
- Radical inverse function
  \[
i = \sum_j a_j(i) b^j \\
\Phi_b(i) = \sum_j a_j(i) b^{-j-1}\]

Halton

- Sample:
  - Where $b_1, b_2, b_3$ are primes
    \[ x_i = (\Phi_{b_1}(i), \Phi_{b_2}(i), \Phi_{b_3}(i), \ldots) \]
    \[ x_i = (\phi_2(i), \phi_3(i), \phi_5(i), \phi_7(i), \phi_{11}(i), \ldots) \]
  - Discrepancy: $O\left(\frac{(\log N)^d}{N}\right)$
Example: Hammersley

• Say we know what N is ahead of time
• For N samples, a Hammersley point
  – \((i/N, \phi_2(i))\)
• For more dimensions:
  – \(X_i=(i/N, \phi_2(i), \phi_3(i), \phi_5(i), \phi_7(i), \phi_{11}(i), \ldots)\)

Quasi Monte Carlo

• Converges as fast as stratified sampling
  – Does not require knowledge about how many samples will be used

• Using QMC, directions evenly spaced no matter how many samples are used

• Samples properly stratified-> better than pure MC
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: next-event estimation

Path Tracing

Sample hemisphere

1 sample/pixel  16 samples/pixel  256 samples/pixel

• Importance Sampling: compute direct illumination separately!
Direct Illumination

- Paths of length 1 only, between receiver and light source
Next Event Estimation

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

Radiance from light sources + radiance from other surfaces

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

\[ \int_{\Omega_s} f_r \cdot \cos \]

• So … sample direct and indirect with separate MC integration
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

Two forms of the RE

- Hemisphere integration

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{\Omega_z} f_z(\Psi \leftrightarrow \Theta) \cdot L(x \leftarrow \Psi) \cdot \cos \theta_x \cdot d\omega_y \]

- Area integration (over polygons from set A)

\[ L(x \rightarrow \Theta) = L_e(x \rightarrow \Theta) + \int_{A} f_y(\Psi \leftrightarrow \Theta) \cdot L(y \leftarrow \Psi) \cdot \frac{\cos \theta_x \cdot \cos \theta_y}{r_{xy}^2} \cdot V(x, y) \cdot dA_j \]
Direct Illumination

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

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

hemisphere integration area integration

Generating direct paths

- Pick surface points \( y_i \) on light source
- Evaluate direct illumination integral

\[ \langle L(x \rightarrow \Theta) \rangle = \frac{1}{N} \sum_{i=1}^{N} f_s(...) \cdot L(...) \cdot G(x, y_i) \cdot \frac{\text{Vis}(x, y)}{p(y)} \]
Applied to direct illumination

\[ p(y) = \frac{1}{\text{Area}_{\text{source}}} \quad E(x) = \text{Area}_{\text{source}} f_r \frac{\cos \theta_x \cos \theta_z}{r^2_{xy}} \text{Vis}(x, y) \]

More points ...

\[ E(x) = \frac{\text{Area}_{\text{source}} f_r L_{\text{source}}}{N} \sum_{i=1}^{N} \frac{\cos \theta_x \cos \theta_z}{r^2_{xy}} \text{Vis}(x, y_i) \]

© Kavita Bala, Computer Science, Cornell University
Even more points ...

\[ E(x) = \frac{Area_{source} f_s L_{source}}{N} \sum_{i=1}^{N} \cos \theta_i \cos \theta_i \frac{Vis(x, y_i)}{r_{x y_i}^2} \]

Parameters

- How many paths ("shadow-rays")?
  - Total?
  - Per light source? (~intensity, importance, …)

- How to distribute paths within light source?
  - Uniform, Solid angle, area
  - What about light distribution?
Generating direct paths

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

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

Direct Paths: Using Area Form

1 path / source 9 paths / source 36 paths / source
Direct Illumination

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

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

hemisphere integration

area integration

Alternative direct paths

- Shoot paths at random over hemisphere; check if they hit light source
  - paths not used efficiently
  - noise in image
  - might work if light source occupies large portion on hemisphere
Alternative direct paths

- Pick random point on random surface; check if on light source and visible to target point
  - paths not used efficiently
  - noise in image
  - might work for large surface light sources in open spaces
Direct path generators

- Light source sampling
  - $L_e$ non-zero
  - 1 visibility term in estimator

- Hemisphere sampling
  - $L_e$ can be 0
  - No visibility in estimator

- Surface sampling
  - $L_e$ can be 0
  - 1 visibility term in estimator

Direct paths

- Different path generators produce different estimators and different error characteristics
- Direct illumination general algorithm:

```c
compute_radiance (point, direction)
  est_rad = 0;
  for (i=0; i<n; i++)
    p = generate_path;
    est_rad += energy_transfer(p) / probability(p);
  est_rad = est_rad / n;
  return(est_rad);
```
Parameters

- How many paths (“shadow-rays”)?
  - Total?
  - Per light source? (~intensity, importance, …)

- How to distribute paths within light source?
  - Uniform, Solid angle, area
  - What about light distribution?

How to sample direct illumination

- Sampling a single light source

- Sampling for many lights
Estimator for direct lighting

- Pick a point on the light’s surface with pdf $p(y)$

- For $N$ samples, direct light at point $x$ is:

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} f_r L_{source} \frac{\cos \theta_x \cos \theta_{\tilde{y}_i}}{r^2_{xy_i}} \frac{Vis(x, \tilde{y}_i)}{p(\tilde{y}_i)}$$

PDF for sampling light

- Uniform

$$p(y) = \frac{1}{Area_{source}}$$

- Pick a point uniformly over light’s area
  - Can stratify samples

- Estimator:

$$E(x) = \frac{Area_{source}}{N} \sum_{i=1}^{N} f_r L_{source} \frac{\cos \theta_x \cos \theta_{\tilde{y}_i}}{r^2_{xy_i}} \frac{Vis(x, \tilde{y}_i)}{p(\tilde{y}_i)}$$
More points ...

\[ E(x) = \frac{Area_{source}}{N} \sum_{i=1}^{N} f_i L_{source} \frac{\cos \theta_x \cos \theta_y}{r^2_{xy}} Vis(x, y_i) \]

Even more points ...

\[ E(x) = \frac{Area_{source}}{N} \sum_{i=1}^{N} f_i L_{source} \frac{\cos \theta_x \cos \theta_y}{r^2_{xy}} Vis(x, y_i) \]
Different pdfs

- Solid angle sampling
  \[ p(y) = \frac{\cos \theta_y}{r^2} \]
  - Removes cosine and distance from integrand
  - Better when significant foreshortening

\[
E(x) = \frac{1}{N} \sum_{i=1}^{N} f_r L_{source} \frac{\cos \theta_x \cos \theta_{y_i}}{r_{xy_i}^2} \frac{Vis(x, \bar{y}_i)}{p(\bar{y}_i)}
\]

Parameters

- Multiple lights
  - Uniform
  - Proportional to power
  - Proportional to area
Strategies for picking light

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

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

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

Parameters

- Multiple lights
  - Uniform
  - Proportional to power
  - Proportional to area

\[
E(x) = \frac{1}{N} \sum_{l=1}^{N} \frac{f_r L_{source} \cos \theta_x \cos \theta_y \cdotVis(x,y_i)}{\rho_{xy}^2} \cdot \frac{1}{p_L(k)p(y_i|k_i)}
\]
Scenes with many lights

- Many lights in scenes: M lights

- How to handle many lights?

- Formulation 1: M integrals, one per light
  – Same solution technique as earlier for each light

\[
L(x \rightarrow \Theta) = \sum_{i=1}^{M} \int_{A_{\text{source}}} f_s(x, -\Psi \leftrightarrow \Theta) \cdot \mathcal{L}_{\text{source}}(y \rightarrow -\Psi') \cdot G(x, y) \cdot dA_y
\]
Scenes with many lights

- Various choices:
  - Shadow rays per light source
  - Distribution of shadow rays within a light source

- Total # rays = M * N
  - Where, M = #lights, N = #rays per source

Antialiasing: pixel

- Anti-aliasing: k * M * N
Formulation over all lights

- When $M$ is large, each direct lighting sample is very expensive
- We would like to importance sample the lights
- Instead of $M$ integrals

$$L(x \rightarrow \Theta) = \sum_{i=1}^{M} \int_{A_{\text{source}}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L_{\text{source}}(y \rightarrow -\Psi) \cdot G(x, y) \cdot dA$$

- Formulation over 1 integration domain

$$L(x \rightarrow \Theta) = \int_{A_{\text{all lights}}} f_r(x, -\Psi \leftrightarrow \Theta) \cdot L_{\text{source}}(y \rightarrow -\Psi) \cdot G(x, y) \cdot dA$$

Why?

- Do not need a minimum of $M$ rays/sample
- Can use only one ray/sample
- Still need $N$ samples, but 1 ray/sample
- Ray is distributed over the whole integration domain
  - Can importance sample the lights
Anti-aliasing

- Can piggy-back on the anti-aliasing of pixel

### How to sample the lights?

- A discrete pdf $p_L(k_i)$ picks the light $k_i$

- A surface point is then picked with pdf $p(y_i|k_i)$

- Estimator with $N$ samples:

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{f_r L_{source} G(x, \bar{y}_i)}{p_L(k_i) p(y_i|k_i)}$$
Strategies for picking light

- Uniform \( p_L(k) = \frac{1}{M} \)
- Area \( p_L(k) = \frac{A_k}{\sum A_k} \)
- Power \( p_L(k) = \frac{P_k}{\sum P_k} \)

Example for 2 lights

- Light 0 has power 1, Light 1 has power 2

- Using power for pdf:
  - \( p_L(L_0) = 1/3 \), \( p_L(L_1) = 2/3 \)

\[
\begin{array}{c|c}
L0 & L1 \\
\hline
0.33 & \\
\end{array}
\]

- Overall pdf
  \[
p(y) = \frac{1}{3} p_{L_0}(y) + \frac{2}{3} p_{L_1}(y)
\]
Example for 2 lights

- Pick a random value: $\xi_0$

- If $\xi_0 < \frac{1}{3}$

- Sample Light 0 and compute estimate $e_0$

- Overall estimate is $\frac{e_0}{\frac{1}{3}}$

Example for 2 lights

- If $\frac{1}{3} \leq \xi_0 < 1$

- Sample Light 1 and compute estimate $e_1$

- Overall estimate is $\frac{e_1}{\frac{2}{3}}$
How to sample light?

- Once light is picked, can pick two random numbers \( \xi_1, \xi_2 \) according to \( p_{L_0}(y), p_{L_1}(y) \)

- To decrease variance we should reuse \( \xi_0 \)

- But, already used information in \( \xi_0 \) to pick the light

Example for 2 lights

- Rescale \( \xi_0 \)
  \[
  \xi'_0 = \frac{3}{2}(\xi_0 - \frac{1}{3})
  \]

\[
(0.533 - 0.333)3/2 = 0.3
\]

- Use \( (\xi'_0, \xi_1) \) to pick samples on light 1
Strategies for picking light

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

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

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

Don’t take visibility into account

---

Research on many lights

• Ward ‘91

• Shirley, Wang, Zimmerman ‘94

• Fernandez, Bala, Greenberg ‘02

• Wald and Slusallek ’03

• Walter et al. ‘05