Monte Carlo Integration for Image Synthesis

Thomas Funkhouser
Princeton University
COS 526, Fall 2002

Main Sources

• Books
  - Realistic Ray Tracing, Peter Shirley
  - Realistic Image Synthesis Using Photon Mapping, Henrik Wann Jensen

• Theses
  - Robust Monte Carlo Methods for Light Transport Simulation, Eric Veach
  - Mathematical Models and Monte Carlo Methods for Physically Based Rendering, Eric La Fortune

• Course Notes
Outline

• Motivation
• Monte Carlo integration
• Monte Carlo path tracing
• Variance reduction techniques
• Sampling techniques
• Conclusion

Motivation

• Rendering = integration
  ○ Antialiasing
  ○ Soft shadows
  ○ Indirect illumination
  ○ Caustics
Motivation

- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

\[ L_p = \int_S L(x \rightarrow e) \, dA \]

Motivation

- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

\[ L(x, \bar{w}) = L_s(x, x \rightarrow e) + \int_S f_r(x, x' \rightarrow x, x \rightarrow e) L(x' \rightarrow x) V(x, x') G(x, x') \, dA \]
Motivation

• Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

\[
L(x, \bar{w}) = L_e(x, x \rightarrow e) + \int f_e(x, x' \rightarrow x, x \rightarrow e) L(x' \rightarrow x)V(x, x')G(x, x')dA
\]

Motivation

• Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

\[
L_e(x, \bar{w}) = L_e(x, \bar{w}) + \int f_e(x, \bar{w}', \bar{w}) L(x, \bar{w}') (\bar{w}' \cdot \bar{n}) d\bar{w}
\]
Motivation

• Rendering = integration
  • Antialiasing
  • Soft shadows
  • Indirect illumination
  • Caustics

\[ L_e(x, \vec{w}) = L_o(x, \vec{w}) + \int_{\Omega} f_r(x, \vec{w}', \vec{w}) L_o(x, \vec{w}') (\vec{w}' \cdot \vec{n}) d\vec{w} \]

Motivation

• Rendering = integration
  • Antialiasing
  • Soft shadows
  • Indirect illumination
  • Caustics

\[ L_e(x, \vec{w}) = L_o(x, \vec{w}) + \int_{\Omega} f_r(x, \vec{w}', \vec{w}) L_o(x, \vec{w}') (\vec{w}' \cdot \vec{n}) d\vec{w} \]
Motivation

- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

\[ L_e(x, \vec{w}) = L_e(x, \vec{w}) + \int_\Omega f_r(x, \vec{w}', \vec{w}) L_e(x, \vec{w}') (\vec{w} \cdot \vec{n}) d\vec{w} \]

Challenge

- Rendering integrals are difficult to evaluate
  - Multiple dimensions
  - Discontinuities
    - Partial occluders
    - Highlights
    - Caustics

\[ L(x, \vec{w}) = L_e(x, x \rightarrow e) + \int_S f_r(x, x' \rightarrow x, x \rightarrow e) L(x' \rightarrow x) V(x, x') G(x, x') dA \]
Challenge

- Rendering integrals are difficult to evaluate
  - Multiple dimensions
  - Discontinuities
    - Partial occluders
    - Highlights
    - Caustics

\[
L(x, \tilde{w}) = L_e(x, x \to e) + \int_s f_r(x, x' \to x, x \to e) L(x' \to x) V(x, x') G(x, x') dA
\]

Outline

- Motivation
- Monte Carlo integration
- Monte Carlo path tracing
- Variance reduction techniques
- Sampling techniques
- Conclusion
Integration in 1D

\[ \int_{0}^{1} f(x) \, dx = ? \]

We can approximate

\[ \int_{0}^{1} f(x) \, dx = \int_{0}^{1} g(x) \, dx \]
Or we can average

\[ \int_0^1 f(x)\,dx = E(f(x)) \]

Estimating the average

\[ \int_0^1 f(x)\,dx = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \]
Other Domains

\[ \int_{a}^{b} f(x) \, dx = \frac{b-a}{N} \sum_{i=1}^{N} f(x_i) \]

\[ \langle f \rangle_{ab} \]

Multidimensional Domains

- Same ideas apply for integration over …
  - Pixel areas
  - Surfaces
  - Projected areas
  - Directions
  - Camera apertures
  - Time
  - Paths

\[ \int_{UGLY} f(x) \, dx = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \]
Outline

- Motivation
- Monte Carlo integration
- Monte Carlo path tracing
- Variance reduction techniques
- Sampling techniques
- Conclusion

Monte Carlo Path Tracing

- Integrate radiance for each pixel by sampling paths randomly

\[ L(x, \tilde{w}) = L_s(x, \tilde{w}) + \int_{\Omega} f_r(x, \tilde{w}', \tilde{w}) L_r(x, \tilde{w}') (\tilde{w}' \cdot \tilde{n}) d\tilde{w}' \]
Simple Monte Carlo Path Tracer

• Step 1: Choose a ray \((x, y), (u, v), t\); weight = 1

• Step 2: Trace ray to find intersection with nearest surface

• Step 3: Randomly decide whether to compute emitted or reflected light
  
  • Step 3a: If emitted,
    return weight * \(L_e\)
  
  • Step 3b: If reflected,
    weight *= reflectance
    Generate ray in random direction
    Go to step 2

Monte Carlo Path Tracing

• Advantages
  
  ◦ Any type of geometry (procedural, curved, ...)
  ◦ Any type of BRDF (specular, glossy, diffuse, ...)
  ◦ Samples all types of paths \((L(SD)\times E)\)
  ◦ Accuracy controlled at pixel level
  ◦ Low memory consumption
  ◦ Unbiased - error appears as noise in final image

• Disadvantages
  
  ◦ Slow convergence
  ◦ Noise in final image
Monte Carlo Path Tracing

Big diffuse light source, 20 minutes

Monter Carlo Path Tracing

1000 paths/pixel
Variance

$$Var[E(f(x))]=\sum_{i=1}^{N}[f(x_i)-E(f(x))]^2$$

Variance decreases with 1/N
Error decreases with 1/sqrt(N)
Outline

- Motivation
- Monte Carlo integration
- Monte Carlo path tracing
- Variance reduction techniques
- Sampling techniques
- Conclusion

Variance

- Problem: variance decreases with $1/N$

More samples removes noise SLOWLY
Variance Reduction Techniques

- Importance sampling
- Stratified sampling
- Metropolis sampling
- Quasi-random

\[ \int_{0}^{1} f(x)dx = \frac{1}{N} \sum_{i=1}^{N} f(x_i) \]

Importance Sampling

- Put more samples where \( f(x) \) is bigger

\[ \int_{\Omega} f(x)dx = \frac{1}{N} \sum_{i=1}^{N} Y_i \]

\[ Y_i = \frac{f(x_i)}{p(x_i)} \]
### Importance Sampling

- This is still “unbiased”

\[
E[Y_i] = \int_{\Omega} Y(x) p(x) dx
\]

\[
= \int_{\Omega} \frac{f(x)}{p(x)} p(x) dx
\]

\[
= \int_{\Omega} f(x) dx
\]

for all \( N \)

### Importance Sampling

- Zero variance if \( p(x) \sim f(x) \)

\[
p(x) = c f(x)
\]

\[
Y_i = \frac{f(x_i)}{p(x_i)} = \frac{1}{c}
\]

\[
Var(Y) = 0
\]

Less variance with better importance sampling
Stratified Sampling

• Estimate subdomains separately

\[ E_k(f(x)) \]

\[ x_1 \quad x_N \]

Stratified Sampling

• This is still unbiased

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

\[ = \frac{1}{N} \sum_{k=1}^{M} N_i F_i \]
Stratified Sampling

- Less overall variance if less variance in subdomains

$$\text{Var}[F_N] = \frac{1}{N^2} \sum_{k=1}^{N} N_i \text{Var}[F_i]$$

Outline

- Motivation
- Monte Carlo integration
- Monte Carlo path tracing
- Variance reduction techniques
- Sampling techniques
- Conclusion
Basic Monte Carlo Path Tracer

- **Step 1:** Choose a ray \((x, y), (u, v), t\)
- **Step 2:** Trace ray to find intersection with nearest surface
- **Step 3:** Randomly decide whether to compute emitted or reflected light
  - **Step 3a:** If emitted, return weight \(\times L_e\)
  - **Step 3b:** If reflected,
    - weight \(\times = \) reflectance
    - Generate ray in random direction
    - Go to step 2

Sampling Techniques

- Problem: how do we generate random points/directions during path tracing?
  - Non-rectilinear domains
  - Importance (BRDF)
  - Stratified
Generating Random Points

- Uniform distribution:
  - Use random number generator

- Specific probability distribution:
  - Function inversion
  - Rejection
  - Metropolis
Generating Random Points

- Specific probability distribution:
  - Function inversion
  - Rejection
  - Metropolis
Generating Random Points

• Specific probability distribution:
  - Function inversion
  - Rejection
  - Metropolis
Generating Random Points

- Specific probability distribution:
  - Function inversion
  - Rejection
  - Metropolis

[Diagram of cumulative probability distribution with x-axis labeled as \( \Omega \) and y-axis labeled as Cumulative Probability, showing a cumulative distribution function graph.]
Generating Random Points

- Specific probability distribution:
  - Function inversion
  - Rejection
  - Metropolis
Combining Multiple PDFs

- Balance heuristic
  - Use combination of samples generated for each PDF
  - Number of samples for each PDF chosen by weights
  - Near optimal

Monte Carlo Path Tracing Image

2000 samples per pixel, 30 SGIs, 30 hours
Monte Carlo Extensions

- Unbiased
  - Bidirectional path tracing
  - Metropolis light transport

- Biased, but consistent
  - Noise filtering
  - Adaptive sampling
  - Irradiance caching
Monte Carlo Extensions

- Unbiased
  - Bidirectional path tracing
  - Metropolis light transport

- Biased, but consistent
  - Noise filtering
  - Adaptive sampling
  - Irradiance caching
Monte Carlo Extensions

• Unbiased
  ◦ Bidirectional path tracing
  ◦ Metropolis light transport

• Biased, but consistent
  ◦ Noise filtering
  ◦ Adaptive sampling
  ◦ Irradiance caching

Monte Carlo Extensions

• Unbiased
  ◦ Bidirectional path tracing
  ◦ Metropolis light transport

• Biased, but consistent
  ◦ Noise filtering
  ◦ Adaptive sampling
  ◦ Irradiance caching
Summary

• Monte Carlo Integration Methods
  ◦ Very general
  ◦ Good for complex functions with high dimensionality
  ◦ Converge slowly (but error appears as noise)

• Conclusion
  ◦ Preferred method for difficult scenes
  ◦ Noise removal (filtering) and
    irradiance caching (photon maps)
    used in practice

Programming Assignment #1
More Information

• Books
  ◦ Realistic Ray Tracing, Peter Shirley
  ◦ Realistic Image Synthesis Using Photon Mapping, Henrik Wann Jensen

• Theses
  ◦ Robust Monte Carlo Methods for Light Transport Simulation, Eric Veach
  ◦ Mathematical Models and Monte Carlo Methods for Physically Based Rendering, Eric La Fortune

• Course Notes