Sampling and Reconstruction of Visual Appearance: From Denoising to View Synthesis

CSE 274 [Fall 2022], Lecture 3
Ravi Ramamoorthi
http://www.cs.ucsd.edu/~ravir

Motivation: Monte Carlo Rendering
- Key application area for sampling/reconstruction
- Modern methods for denoising now popular
- 1-3 order of magnitude speedups in mature area
- Denoising now standard in production rendering
  - And in real-time, going down to 1spp
- This, next week: Basic background in rendering
  - Reflection and Rendering Equations
  - Monte Carlo Integration
  - Path Tracing (Basic Monte Carlo rendering method)
  - Also the basics of CSE 168 (163)
- Sign up (email me) re paper presentations

Illumination Models
Local Illumination
- Light directly from light sources to surface
- No shadows (cast shadows are a global effect)

Global Illumination: multiple bounces (indirect light)
- Hard and soft shadows
- Reflections/refractions (already seen in ray tracing)
- Diffuse and glossy interreflections (radiosity, caustics)

Some images courtesy Henrik Wann Jensen

Caustics
Caustics: Focusing through specular surface
- Major research effort in 80s, 90s till today

Overview of lecture
- Theory for all global illumination methods (ray tracing, path tracing, radiosity)
- We derive Rendering Equation [Kajiya 86]
  - Major theoretical development in field
  - Unifying framework for all global illumination
  - Introduced Path Tracing: core rendering method
- Discuss existing approaches as special cases

Outline
- Reflectance Equation
- Global Illumination
- Rendering Equation
- As a general Integral Equation and Operator
- Approximations (Ray Tracing, Radiosity)
- Surface Parameterization (Standard Form)
Reflection Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + \sum L_i(x, \omega_i) f(x, \omega_i, \omega_r)(\omega_i \cdot n) \]

Reflected Light (Output Image)  
Emission  
Incident Light (from light source)  
BRDF  
Cosine of Incident angle

Environment Maps

- Light as a function of direction, from entire environment
- Captured by photographing a chrome steel or mirror sphere
- Accurate only for one point, but distant lighting same at other scene locations (typically use only one env. map)

The Challenge

- Computing reflectance equation requires knowing the incoming radiance from surfaces
- But determining incoming radiance requires knowing the reflected radiance from surfaces

Rendering Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_r) + \int_{\Omega} L_i(x', \omega_i) f(x, \omega_i, \omega_r) \cos \theta d\omega_i \]

Reflected Light (Output Image)  
Emission  
Reflected Light  
BRDF  
Cosine of Incident angle

Surfaces (interreflection)  
\( dA \)  
\( d\omega \)
Outline

- Reflectance Equation (review)
- Global Illumination
- Rendering Equation
- As a general Integral Equation and Operator
- Approximations (Ray Tracing, Radiosity)
- Surface Parameterization (Standard Form)

Rendering Equation (Kajiya 86)

Rendering Equation as Integral Equation

$$l(u) = e(u) + \int l(v) K(u,v) \, dv$$

Is a Fredholm Integral Equation of second kind [extensively studied numerically] with canonical form

$$L_r(x, \omega_r) = L_e(x, \omega_r) + \int K(x, \omega_r, \omega_i) \cos \theta_i \, d\omega_i$$

Kernel of equation

Linear Operator Theory

- Linear operators act on functions like matrices act on vectors or discrete representations
  $$h(u) = (M \circ f)(u)$$
  M is a linear operator, f and h are functions of u
  a and b are scalars
  M \circ (af + bg) = a(M \circ f) + b(M \circ g)

- Basic linearity relations hold
  $$fDf(u) = \frac{\partial f}{\partial u}$$

Examples include integration and differentiation

Solving the Rendering Equation

- Too hard for analytic solution, numerical methods
- Approximations, that compute different terms, accuracies of the rendering equation
- Two basic approaches are ray tracing, radiosity. More formally, Monte Carlo and Finite Element. Today Monte Carlo path tracing is core rendering method
- Monte Carlo techniques sample light paths, form statistical estimate (example, path tracing)
- Finite Element methods discretize to matrix equation

Linear Operator Equation

$$l(u) = e(u) + \int l(v) K(u,v) \, dv$$

Kernel of equation

Light Transport Operator

$$L = E + KL$$

Can be discretized to a simple matrix equation [or system of simultaneous linear equations]
(L, E are vectors, K is the light transport matrix)
Solving the Rendering Equation

- General linear operator solution. Within raytracing:
- General class numerical Monte Carlo methods
- Approximate set of all paths of light in scene

$$L = E + KL$$

$$IL - KL = E$$

$$(I - K)L = E$$

$$L = (I - K)^{-1}E$$

Binomial Theorem

$$L = (I + K + K^2 + K^3 + \ldots)E$$

$$L = E + KE + K^2E + K^3E + \ldots$$

Term n corresponds to n bounces of light

Ray Tracing

$$L = E + KE + K^2E + K^3E + \ldots$$

Emission directly

From light sources

Direct Illumination

on surfaces

Global Illumination

(One bounce indirect)

[Mirrors, Reflection]

(Two bounce indirect)

[Caustics etc]

OpenGL Shading

Emission directly

From light sources

Direct Illumination

on surfaces

Global Illumination

(One bounce indirect)

[Mirrors, Reflection]

(Two bounce indirect)

[Caustics etc]

Successive Approximation

Rendering Equation

$$L_x(\omega) = L_x(\omega_x) + \int_{\Omega} L_y(\omega - \omega_x) f(\omega_x, \omega_y) \cos \theta d\omega_y$$

Reflecting Light

(Emission)

[Unknown]

Relected Light

[BRDF]

Cosine of

Incident angle

[Known]

Surface Parameterization (Standard Form)

Outline

- Reflectance Equation (review)
- Global Illumination
- Rendering Equation
- As a general integral Equation and Operator
- Approximations (Ray Tracing, Radiosity)
- Surface Parameterization (Standard Form)
Change of Variables

\[ L(x, \omega_i) = L(x, \omega_i) + \int_{\omega} L(x', -\omega_i) f(x, \omega_i, \omega) \cos \theta \, d\omega \]

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)

\[ d\omega = \frac{dA \cos \theta}{|x-x'|^2} \]

Rendering Equation: Standard Form

\[ L(x, \omega_i) = L(x, \omega_i) + \int_{\omega} L(x', -\omega_i) f(x, \omega_i, \omega) \cos \theta \cos \theta \, d\omega \]

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)

\[ d\omega = \frac{dA \cos \theta \cos \theta}{|x-x'|^2} \]

\[ G(x, x') = G(x', x) = \frac{\cos \theta \cos \theta}{|x-x'|^2} \]

Summary

- **Theory** for all global illumination methods (ray tracing, path tracing, radiosity)
- We derive **Rendering Equation** [Kajiya 86]
- Major theoretical development in field
- Unifying framework for all global illumination
- Discuss existing approaches as special cases

Motivation: Monte Carlo Integration

Rendering = integration
- Reflectance equation: Integrate over incident illumination
- Rendering equation: Integral equation

Many sophisticated shading effects involve integrals
- Anti-aliasing
- Soft shadows
- Indirect illumination
- Caustics

Most Sampling/Reconstruction treats actual rendering as a black box. But still helpful to know some basics

Example: Soft Shadows

\[ E(x) = \int_{\Omega} L_i(x, \omega) \cos \theta \, d\omega \]

Challenges
- Visibility and blockers
- Varying light distribution
- Complex source geometry

Source: Agrawala, Ramamoorthi, Heirich, Moll, 2000
Monte Carlo

- Algorithms based on statistical sampling and random numbers
- Coined in the beginning of 1940s. Originally used for neutron transport, nuclear simulations
  - Von Neumann, Ulam, Metropolis, …
- Canonical example: 1D integral done numerically
  - Choose a set of random points to evaluate function, and then average (expectation or statistical average)

Monte Carlo Algorithms

Advantages
- Robust for complex integrals in computer graphics (irregular domains, shadow discontinuities, and so on)
- Efficient for high dimensional integrals (common in graphics: time, light source directions, and so on)
- Quite simple to implement
- Work for general scenes, surfaces
- Easy to reason about (but care taken re statistical bias)

Disadvantages
- Noisy
- Slow (many samples needed for convergence)
- Not used if alternative analytic approaches exist (but those are rare)

Integration in 1D

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

We can approximate

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

Standard integration methods like trapezoidal rule and Simpson's rule

Advantages:
- Converges fast for smooth integrands
- Deterministic

Disadvantages:
- Exponential complexity in many dimensions
- Not rapid convergence for discontinuities

Monte Carlo methods (random choose samples)

Advantages:
- Robust for discontinuities
- Converges reasonably for large dimensions
- Can handle complex geometry, integrals
- Relatively simple to implement, reason about

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

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

Multidimensional Domains

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

Random Variables

- Describes possible outcomes of an experiment
- In discrete case, e.g. value of a dice roll \([x = 1-6]\)
- Probability \(p\) associated with each \(x\) (1/6 for dice)
- Continuous case is obvious extension

Expected Value

- Expectation
  - Discrete: \(E(f) = \sum_{i=1}^N p_i f(x_i)\)
  - Continuous: \(E(f) = \int_0^1 p(x) f(x) \, dx\)

For Dice example:

\[
E(x) = \sum_{i=1}^6 \frac{1}{6} x_i = \frac{1}{6}(1+2+3+4+5+6) = 3.5
\]

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

Common Operations

Want to sample probability distributions
- Draw samples distributed according to probability
- Useful for integration, picking important regions, etc.

Common distributions
- Disk or circle
- Uniform
- Upper hemisphere for visibility
- Area luminaire
- Complex lighting like an environment map
- Complex reflectance like a BRDF

Sampling Continuous Distributions

Cumulative probability distribution function
\[ P(x) = \Pr(X < x) \]

Construction of samples
Solve for \( X = F^{-1}(U) \)

Must know:
1. The integral of \( p(x) \)
2. The inverse function \( F^{-1}(x) \)

Example: Power Function

Assume
\[ p(x) = \frac{(n+1)x^n}{\int_0^1 x^n dx} = \frac{1}{n+1} \]

\[ P(x) = x^{n+1} \]

\[ X \sim p(x) \Rightarrow X = F^{-1}(U) = \sqrt{U} \]

Trick
\[ Y = \max(U_1, U_2, \ldots, U_k, U_{n+1}) \]

\[ \Pr(Y < x) = \prod_{i=1}^{n+1} \Pr(U < x) = x^{n+1} \]
Sampling a Circle

\[ A = \int_0^{\frac{\pi}{2}} r \, dr \, d\theta = \int_0^{\frac{\pi}{2}} r \, dr \cdot \left( \frac{r^2}{2} \right) = \pi \]

\[ p(r, \theta) \, dr \, d\theta = \frac{1}{\pi} \, r \, dr \, d\theta \Rightarrow p(r, \theta) = \frac{r}{\pi} \]

\[ p(\theta) = \frac{1}{2\pi} \]

\[ p(r) = \frac{1}{2\pi} r \]

\[ \theta = 2\pi U_1 \]

\[ r = \sqrt{U_2} \]

Rejection Sampling

\[ I = \int_0^1 f(x) \, dx = \int_{y = f(x)} dx \, dy \]

Algorithm

1. Pick \( U_1 \) and \( U_2 \)
2. Accept \( U_1 \) if \( U_2 < f(U_1) \)

Wasteful? Efficiency = Area / Area of rectangle

Sampling a Circle: Rejection

May be used to pick random 2D directions

Circle techniques may also be applied to the sphere

More formally

Definite integral

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

Expectation of \( f \)

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

Random variables

\[ X_i \sim p(x) \]

\[ Y_i = f(X_i) \]

Estimator

\[ F_N = \frac{1}{N} \sum_{i=1}^N Y_i \]
**Unbiased Estimator**

\[ E[F_N] = I(f) \]

\[ E[F_N] = \frac{1}{N} \sum_{i=1}^{N} Y_i \]

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

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

**Properties**

\[ E[\sum Y_i] = \sum E[Y_i] \]

\[ E[aY] = aE[Y] \]

Assume uniform probability distribution for now

**Direct Lighting – Directional Sampling**

\[ E(x) = \int_{\Omega} L(x, \omega) \cos \theta \, d\omega \]

Ray intersection \( x'(x, \omega) \)

Sample \( \omega \) uniformly by \( \Omega \)

\[ Y_i = L(x'(x, \omega), -\omega) \cos \theta \]

for all \( N \)

**Importance Sampling**

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

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

\[ Y_i = \frac{f(x)}{p(x)} \]

**Importance Sampling**

This is still unbiased

\[ E[Y_1] = \int_{\Omega} Y(x) p(x) \, dx \]

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

for all \( N \)

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

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

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

\[ \text{Var}(Y) = 0 \]

Less variance with better importance sampling
Stratified Sampling

- Estimate subdomains separately

\[ E(f(x)) \]

\[ x_1 \]

\[ x_N \]

\[ \sum_{k=1}^{M} \frac{1}{N} \sum_{i=1}^{N} \text{Var}(F_k) \]

61

Stratified Sampling

- Less overall variance if less variance in subdomains

\[ \text{Var}[F_i] = \frac{1}{N} \sum_{k=1}^{M} N \text{Var}[F_k] \]

62

More Information

- Veach PhD thesis chapter (linked to from website)
- Course Notes (links from website)

63