To Do

- Assignment 2 due May 15
  - Should already be well on way.
  - Contact us for difficulties etc.

This lecture on rendering, rendering equation. Pretty advanced theoretical material. Don’t worry if a bit lost; not directly required on the homeworks.

Course Outline

- 3D Graphics Pipeline
  - Modeling (Creating 3D Geometry)
  - Rendering (Creating, shading images from geometry, lighting, materials)

Unit 1: Foundations of Signal and Image Processing
Understanding the way 2D images are formed and displayed, the important concepts and algorithms, and to build an image processing utility like Photoshop
Weeks 1 – 3. Assignment 1

Unit 2: Meshes, Modeling
Weeks 3 – 5. Assignment 2

Unit 3: Advanced Rendering
Weeks 6 – 8. (Final Project)

Unit 4: Animation, Imaging
Weeks 9, 10. (Final Project)

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)

Diffuse Interreflection
Diffuse interreflection, color bleeding [Cornell Box]
Radiosity

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
  - Discuss existing approaches as special cases

Fairly theoretical lecture (but important). Not well covered in textbooks (though see Eric Veach’s thesis). See reading if you are interested.

Outline

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

Outline

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

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
Reflection Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_i) + \int \omega_r L_i(x, \omega_i) f(x, \omega_i, \omega_r) \cos \theta \, d\omega_i \]

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)

Environment Maps

- Environment maps widely used as lighting representation
- Many modern methods deal with offline and real-time rendering with environment maps
- Image-based complex lighting + complex BRDFs

The Challenge

\[ L_i(x, \omega_i) = L_e(x, \omega_i) + \int \Omega \int L_i(x', \omega_i) f(x, \omega_i, \omega_r) \cos \theta \, d\omega_i \]

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

Rendering Equation

\[ L_r(x, \omega_r) = L_e(x, \omega_i) + \int \omega_r L_i(x, \omega_i) f(x, \omega_i, \omega_r) \cos \theta \, d\omega_i \]

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

Rendering Equation (Kajiya 86)

- Surfaces (interreflection)
- Emission
- Reflected Light
- BRDF
- Cosine of Incident angle

UNKNOWN UNKNOWN KNOWN KNOWN KNOWN

Figure 8: A sample image. All objects are casting gray, green, and black shadows due to lighting in the scene.
Rendering Equation as Integral Equation

\[ L(x, \omega_r) = I(x, \omega_r) + \int L(x', -\omega_r) \cdot \text{BRDF} \cdot \cos(\theta_{\omega_i}) \, dv \]

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

\[ I(u) = e(u) + \int K(u, v) \, dv \]

Kernel of equation

Linear Operator Equation

\[ I(u) = e(u) + \int K(u, v) \, dv \]

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)

Ray Tracing and extensions

- General class numerical Monte Carlo methods
- Approximate set of all paths of light in scene

\[ L = E + KL \]

\[ (I - K) L = E \]

\[ L = (I - K)^{-1} E \]

Binomial Theorem

\[ L = (I + K + K^2 + ... E) \]

\[ L = E + KE + K^2 E + K^3 E + ... \]

Ray Tracing

\[ L = E + KE + K^2 E + K^3 E + ... \]

Emission directly From light sources

Direct Illumination on surfaces

Global Illumination (One bounce indirect) [Mirrors, Refraction]

[Two bounce indirect] [Caustics etc]

Outline

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

OpenGL Shading

Direct Illumination on surfaces

Global Illumination (One bounce indirect) [Mirrors, Refraction]

[Two bounce indirect] [Caustics etc]
Rendering Equation as Integral Equation

\[
L(x,u) = L_0(x,u) + \int L(x',u') f(x',u') \cos\theta du' \]

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

\[
I(u) = e(u) + \int l(v) K(u,v) dv
\]

Linear Operator Theory

• Linear operators act on functions like matrices act on vectors or discrete representations

\[
h(u) = (M \circ f)(u)
\]

• Basic linearity relations hold \( a \) and \( b \) are scalars \( f \) and \( g \) are functions

\[
M(a f + b g) = a(M f) + b(M g)
\]

• Examples include integration and differentiation

\[
(K \circ f)(u) = \int k(u,v) f(v) dv
\]

\[
(D \circ f)(u) = \frac{df}{du}(u)
\]

Solving the Rendering Equation

\[
L = E + KL
\]

\[
I = E
\]

\[
I - KL = E
\]

\[
L = (I - K)^{-1} E
\]

Binomial Theorem

\[
L = (I + K + K^2 + K^3 + ...) E
\]

\[
L = E + KE + K^2 E + K^3 E + ...
\]

Term \( n \) corresponds to \( n \) bounces of light

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
• Monte Carlo techniques sample light paths, form statistical estimate (example, path tracing)
• Finite Element methods discretize to matrix equation

Ray Tracing

\[
L = E + KE + K^2 E + K^3 E + ...
\]

Emission directly

From light sources

Direct Illumination on surfaces

Global Illumination (One bounce indirect) [Mirrors, Refraction]

(Two bounce indirect [Caustics etc]
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, refraction)
- (Two bounce indirect, caustics etc)

OpenGL Shading

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

\[ L_\omega(x) = L_\omega(x) + \int \frac{\cos \theta d\omega}{|x-x'|} \]

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

Rendering Equation

\[ L_r(x, \omega) = L_r(x, \omega) + \int L_r(x', \omega') \gamma(x, \omega, \omega') \cos \theta d\omega \]

Reflected Light (Output Image)
UnKnown

Reflective Light
Known

BRDF
UnKnown

Cosine of incident angle
Known

Change of Variables

\[ L_r(x, \omega) = L_r(x, \omega) + \int L_r(x', \omega) \gamma(x, \omega, \omega') \cos \theta d\omega \]

Integral over angles sometimes insufficient. Write integral in terms of surface radiance only (change of variables)
Render Equation: Standard Form

\[ L(x, \omega_i) = L_e(x, \omega_i) + \int L_s(x', -\omega_i) I(x, \omega_i, \omega_o) \cos \theta_i d\omega_o \]

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

\[ L(x, \omega_i) = L(x, \omega_i) + \int L_s(x', -\omega_i) I(x', \omega_i, \omega_o) \cos \theta_i d\omega_o \]

Domain integral awkward. Introduce binary visibility fn \( V \)

\[ \Omega' = \frac{2 \cos \omega_i |x - x'|}{|x - x'|^2} \]

Rendering Equation

\[ L(x, \omega_i) = L(x, \omega_i) + \int L_s(x', -\omega_i) I(x, \omega_i, \omega_o) G(x, x') V(x, x') dA' \]

Radiosity Equation

\[ L(x, \omega_i) = L(x, \omega_i) + \int L_s(x', -\omega_i) I(x, \omega_i, \omega_o) G(x, x') V(x, x') dA' \]

Discretization and Form Factors

\[ B_i = E_i + \rho \sum_j B_j F_{j\rightarrow i} \frac{A_j}{A_i} \]

\( F \) is the form factor. It is dimensionless and is the fraction of energy leaving the entirety of patch \( j \) (multiply by area of \( j \) to get total energy) that arrives anywhere in the entirety of patch \( i \) (divide by area of \( i \) to get energy per unit area or radiosity).

\[ B_i = E_i + \rho \sum_j B_j F_{j\rightarrow i} \]

Form Factors

\[ A F_{i\rightarrow j} = A_i F_{j\rightarrow i} = \int \frac{G(x, x') V(x, x') dA_i dA_j}{\pi} \]

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

Matrix Equation

\[ B_i = E_i + \rho \sum_j B_j F_{j\rightarrow i} \frac{A_j}{A_i} \]

\[ A F_{i\rightarrow j} = A_i F_{j\rightarrow i} = \int \frac{G(x, x') V(x, x') dA_i dA_j}{\pi} \]

\[ B_i = E_i + \rho \sum_j B_j F_{j\rightarrow i} \]

\[ \sum_j M_j B_j = E_i \quad MB = E \quad M_j = I_j - \rho F_{i\rightarrow j} \]

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
- Next: Practical solution using Monte Carlo methods