Global Illumination

Greg Humphreys
CS445: Intro Graphics
University of Virginia, Fall 2003

Lighting Simulation

- The Rendering Equation
  Given a scene consisting of geometric primitives with material properties and a set of light sources, compute the illumination at each point on each surface
- Challenges
  - Primitives complex: lights, materials, shapes
  - Exponential number of paths, dense coupling
- How to solve it?
  - Radiosity  Finite element
  - Ray Tracing  Monte Carlo

Lighting Example: Cornell Box

- Caustics
- Surface Color
- Incorrect illumination

Lighting Example: Diffuse Reflection

- Surface Color
- Diffuse Shading
<table>
<thead>
<tr>
<th>Lighting Example: Shadows</th>
<th>Lighting Example: Soft Shadows</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Shadows</td>
<td>Hard Shadows Point Light Source</td>
</tr>
<tr>
<td>Shadows</td>
<td>Soft Shadows Area Light Source</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Radiosity: Cornell Experment</th>
<th>Early Radiosity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured</td>
<td>Program of Computer Graphics Cornell University</td>
</tr>
<tr>
<td>Simulated</td>
<td></td>
</tr>
</tbody>
</table>
Very Early Radiosity

Parry Moon and Domina Spencer *Lighting Design* (1948 - MIT)

Lighting Effects

- Hard Shadows
- Soft Shadows
- Caustics
- Indirect Illumination

Caustics

Henrik Wann Jensen 1995

Complex Indirect Illumination

*Courtyard House with Curved Elements*  
*Mies van der Rohe*

Modeling: Stephen Duck; Rendering: Henrik Wann Jensen
Measuring things

- **Flux (Power):** rate at which light energy is emitted
  - Measured in?
- **Solid angle:** 3D generalization of angle
  - Measured in?

- **Intensity:** Flux per solid angle
  - Measured in?

Radiance

- **Radiance:** intensity per unit foreshortened area
  - Foreshortened area found by multiplying the area by \( \cos(\theta) \)
  - Think of the projection of the area onto the plane perpendicular to the direction of radiation

- **Properties of radiance:**
  - Remains constant along a ray
  - The response of a sensor is proportional to the incident radiance

Irradiance

- **Irradiance:** flux per unit area

\[
E = \int_{\Omega} L_i \cos \theta_i \, d\omega_i
\]

Example: Point Light Sources

- Energy distribution has an irritating singularity

- The flux in some small differential solid angle is

\[
d\Phi = I(\vec{\omega}) \, d\omega
\]

- Assume isotropic light source

- Then irradiance at a point on a unit sphere is:

\[
I = \frac{\Phi}{4\pi}
\]
Point Source Irradiance

- Irradiance on a small surface from a point light

\[ E = I \frac{d\omega}{dA} = \frac{\Phi}{4\pi} \cos \theta \]

\( \text{Point Source Irradiance} \)

BRDF's

- Bidirectional Reflection Distribution Function
  - How much light is reflected in direction \( \omega_o \) from direction \( \omega_i \)?

\[ f_r(\theta_i, \phi_i, \theta_o, \phi_o) \]

- Two main properties:
  - Reciprocity: \( f_r(\omega_o \rightarrow \omega_i) = f_r(\omega_i \rightarrow \omega_o) \)
  - Energy preservation: \( \int f_r(\omega_o \rightarrow \omega_i) d\omega \leq 1 \)

The Reflection Equation

- How much light reflects in some given direction?
- Take light coming from all incoming directions, multiply it by the BRDF, multiply by \( \cos(\theta) \)

\[ L_r(\hat{\omega}_r) = \int_{\Omega} f_r(\omega_i \rightarrow \omega_r) L_i(\hat{\omega}_i) \cos \theta_i d\omega_i \]

Aside: Delta Functions

- The Dirac delta function is defined as follows:

\[ \delta(x) = 0 \text{ if } x \neq 0 \]

\[ \int_{-\infty}^{\infty} \delta(x) \, dx = 1 \]

\[ \int_{-\infty}^{\infty} \delta(x - y) \, f(x) \, dx = f(y) \]
### BRDF Example: Mirror

- Mirror reflection, so:
  \[ \theta_r = \theta_i \]
  \[ \phi_r = \phi_i \pm \pi \]
- Also, since no light is absorbed:
  \[ L_r(\theta, \phi) = L_i(\theta, \phi, \pm \pi) \]
- Mirror’s BRDF uses delta functions to enforce this:
  \[ f_r = \frac{\delta(\cos \theta - \cos \theta_i)}{\cos \theta_i} \delta(\phi_i - (\phi_r \pm \pi)) \]

### Diffuse Reflection

- Light is equally likely to be scattered in any direction, regardless of the incident direction
- The BRDF is a constant!
  \[ L_r(\vec{\omega_r}) = \int_{\Omega} f_r L_i(\vec{\omega_i}) \cos \theta_i \omega_i \]
  \[ = f_r \int_{\Omega} L_i(\vec{\omega_i}) \cos \theta_i \omega_i \]
  \[ = f_r E \]

### Indirect Illumination

- Radiance is invariant along a ray
- The radiance at \( x' \) due to the radiance from \( x \) is
  \[ L_i(x', \omega_i) = L_o(x, \omega_o) V(x, x') \]
- \( V(x, x') \) is a boolean \textit{visibility} function

### So... Close...

- Hemispherical integral bad. Surface integral good.
- Relationship between solid angle and projected surface area:
  \[ d\omega = \frac{\cos \theta_0 \, dA}{||x-x'||^2} \]
- So define \( G \):
  \[ G(x, x') = \frac{\cos \theta \cos \theta_0}{||x-x'||} V(x, x') \]
- Change variables in the reflection equation:
  \[ L(x', \omega') = \int_S f_r(x) L(x, \omega) G(x, x') \, dA \]
The Rendering Equation

• Incorporate emission:

\[ L(x', \omega') = L_e(x', \omega') + \int_S f_r(x) L(x, \omega) G(x, x') dA \]

• This completely captures all light transport in a scene
  ◦ Is this true?
• Global illumination = solve the rendering equation
• But it’s too hard!

The Radiosity Equation

• Assume all surfaces are diffuse
• BRDF is a constant, we can pull it out of the integral

\[ B(x) = E(x) + f_r(x) \int_S B(x') G(x, x') dA \]

Solving the radiosity equation

• Radiosity solutions are view-independent!

• This is actually a tractable problem!
• Bounce light around in the scene, absorbing some and reflecting some, until everything settles down
  ◦ What’s that called?

The Radiosity Equation

• We can’t compute integrals, but that’s OK
• Cut up the scene into little “patches”
• Sum the light contribution over all patches:

\[ B_i = E_i + \rho \sum_{j=1}^{N} B_j F_{i \rightarrow j} \frac{A_j}{A_i} \]

• \( F_{i \rightarrow j} \): the fraction of energy leaving patch \( j \) that arrives at patch \( i \).
• Called the form factor between the two patches
Form Factor Facts

- Reciprocity relationship between form factors:
  \[ A_i F_{i \to j} = A_j F_{j \to i} \]

- Simplify the summation:
  - this is pretty simple, considering the rendering equation
  \[ B_i = E_i + \rho_i \sum_{j=1}^{N} B_j F_{i \to j} \]

- Rearrange terms:
  \[ B_i - \rho_i \sum_{j=1}^{N} B_j F_{i \to j} = E_i \]

Wait A Minute…

- This looks suspiciously like a system of equations!

\[
B_i - \rho_i \sum_{j=1}^{N} B_j F_{i \to j} = E_i
\]

- This is Ye Olde Huge Matrixe!
- Solve it using numerical techniques

Finding Form Factors

- The form factor between two tiny surface patches is:
  \[ dF_{i \to j} = \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_y dA_j \]

- \( V_y \) is the binary visibility function

- So the true form factor is:
  \[ F_{i \to j} = \frac{1}{A_j} \int_{A_i} \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{\pi r^2} V_y dA_j dA_i \]

Nusselt’s Method

- Project the visible areas of \( A_i \) onto a unit hemisphere centered at \( dA_i \), and then onto the base
- The ratio of this projected area to the area of the base circle is the form factor
Hemicubes

- Approximate the hemisphere:

![Hemicube Diagram](image)

- Each small hemicube cell has a precomputed delta form factor:

\[ \Delta F_p = \frac{\cos \theta \cos \theta}{\pi r^2} \Delta A \]

- We can render the scene using normal Z-buffer scan conversion onto the faces of the hemicube!

Progressive Refinement

- Radiosity solving is really slow

- Display a reasonable picture while solving

- The approach so far:
  - estimate the radiosity of patch \( l \) based on the estimates of all other patch radiosities:

\[
B_i \text{ due to } B_j = \rho_j B_j F_{i\rightarrow j}
\]

- This is called gathering

- Poor intermediate results
  - Why?

Shooting

- Instead of gathering light, we can shoot the light energy stored at each patch to every other patch:

\[
B_j \text{ due to } B_i = \rho_j B_j F_{i\rightarrow j}
\]

- This requires knowing all the form factors at once
  - That’s bad

- Rewrite the equation as:

\[
B_j \text{ due to } B_i = \rho_j B_j F_{i\rightarrow j} \frac{A_j}{A_i}
\]

- Choose the patch with maximum stored energy
  - Starting with the light sources
Intermediate Results

- Display the latest radiosity values at each patch
- Use ambient to make up the difference
- Set initial radiosities to the emission
- Compute an average diffuse reflectivity for the scene

\[ \rho_{\text{avg}} = \frac{\sum_{i=1}^{N} \rho_i A_i}{\sum_{i=1}^{N} A_i} \]

Intermediate Results

- Compute an overall reflection factor
  - Take into account all paths which energy can take
  
  \[ R = 1 + \rho_{\text{avg}} + \rho_{\text{avg}}^3 + \rho_{\text{avg}}^3 + \ldots = \frac{1}{1 - \rho_{\text{avg}}} \]

- Weight the unshot radiosity by the ratio of the patch’s area to the total area:

\[ \text{Ambient} = R \frac{\sum_{i=1}^{N} \Delta B_i A_i}{\sum_{i=1}^{N} A_i} \]

Shooting Without Ambient

Shooting With Ambient