Today

- The Rendering Equation
- Radiosity Method
- Photon Mapping
- Ambient Occlusion
The light shining on $x$ from $x'$ is equal to:
- the emitted light from $x'$ toward $x$, plus
- for each bit of surface in the scene, how much light shines from that bit onto $x'$ and is reflected toward $x$, scaled appropriately.
The light shining on \( x \) from \( x' \) is equal to:
- the emitted light from \( x' \) toward \( x \), plus
- for each bit of surface in the scene, how much light shines from that bit onto \( x' \) and is reflected toward \( x \), scaled appropriately

\[
L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x'||^2} dx'' \right]
\]
The Rendering Equation

\[ L_s(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x}, \mathbf{x}') \left[ E(\mathbf{x}, \mathbf{x}') + \int_S \rho_{\mathbf{x}'}(\mathbf{x}, \mathbf{x}'') L_s(\mathbf{x}', \mathbf{x}'') \frac{\cos(\theta') \cos(\theta'')}{||\mathbf{x}' - \mathbf{x}''||^2} d\mathbf{x}'' \right] \]
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'')L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x'||^2} dx'' \right] \]

Light energy hitting \( x \) from \( x' \)
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho(x', x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} \, dx'' \right] \]
The Rendering Equation

\[ L_s(x, x') = \begin{cases} \delta(x, x') & \text{if } x \text{ see } x' \\ E(x, x') + \int_S \rho_x(x, x'') L_s(x', x'') \cos(\theta') \cos(\theta'') \frac{dx''}{||x' - x''||^2} \end{cases} \]
The Rendering Equation

\[
L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta')}{{||x' - x''||}^2} \right]
\]

\[
= \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta')}{{||x' - x''||}^2} \right] dx''
\]
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta')}{|x' - x''|^2} \right] \]

Light emitted from x' toward x

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{|x' - x''|^2} \right] \]

\[ x \]
\[ \hat{n'} \]
\[ \Delta \]
\[ \theta' \]
\[ x' \]
\[ \Delta \hat{n''} \]
\[ \theta'' \]
\[ x'' \]
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} \, dx'' \right] \]
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta')}{||x' - x''||^2} \right] \]

sum over every bit of surface in the scene
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} \, dx'' \right] \]
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho(x') L_s(x', x'') \cos(\theta') \cos(\theta'') \frac{\left| x' - x'' \right|^2}{dx''} \right] \]

Light emitted from \( x'' \) toward \( x' \)
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} \, dx'' \right] \]
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} dx'' \right] \]

scaled down by the BRDF of \( x' \)
The Rendering Equation

\[
L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} dx'' \right]
\]
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} dx'' \right] \]

scaled down by distance and relative orientation ("form factor")
The Rendering Equation

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'')L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} \, dx'' \right] \]
Radiosity

- Assume all materials are perfectly Lambertian (diffuse only, no specularities)
  - Removes all dependance on directions
  - Reduces dimensionality of lightfield
  - Allows a FEM solution (break up into chunks)
- Can also relax assumption slightly...
Early radiosity

from Hanrahan 2000
Assume Lambertian

\[
L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{|x' - x''|^2} dx'' \right]
\]

\[
L_s(x, x') = \delta(x, x') \left[ E_{x'} + \int_S \rho_{x'} L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{|x' - x''|^2} dx'' \right]
\]
Assume Lambertian

\[ L_s(x, x') = \delta(x, x') \left[ E(x, x') + \int_S \rho_{x'}(x, x'') L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} dx'' \right] \]

\[ L_s(x, x') = \delta(x, x') \left[ E_{x'} + \int_S \rho_{x'} L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} dx'' \right] \]

Only term dependent on \( x \)
Rewrite in Terms of Radiosity

\[ L_s(x, x') = \delta(x, x') \left[ E_{x'} + \int_S \rho_{x'} L_s(x', x'') \frac{\cos(\theta') \cos(\theta'')}{||x' - x''||^2} \, dx'' \right] \]

\[ H_{x'} = E_{x'} + \rho_{x'} \int_S \delta(x', x'') \frac{H_{x''} \cos(\theta') \cos(\theta'')}{2\pi} \frac{1}{||x' - x''||^2} \, dx'' \]

Note: we changed defn of \( E \) here.
Discretize into Patches

Piece-wise constant patches

Example mesh for Cornell Box by Mark Schmelzenbach
Discretize into Patches

The Candlestick Theater,
Mark Mack Architects.
Discretize into Patches

The Candlestick Theater,
Mark Mack Architects.
**Rewrite in Terms of Patches**

\[
H_{x'} = E_{x'} + \rho_{x'} \int_S \delta(x', x'' \frac{H_{x''} \cos(\theta') \cos(\theta'')}{2\pi \|x' - x''\|^2} \, dx''
\]

\[
H_i = E_i + \rho_i \sum_j H_j \int_{S_j} \delta_{ij} \frac{\cos(\theta_i) \cos(\theta_j)}{2\pi \|c_i - x\|^2} \, dx
\]
Rewrite in Terms of Patches

\[ H_{x'} = E_{x'} + \rho_{x'} \int_S \delta(x', x'') \frac{H_{x''} \cos(\theta') \cos(\theta'')}{2\pi \|x' - x''\|^2} \, dx'' \]

\[ H_i = E_i + \rho_i \sum_j H_j \int_{S_j} \frac{\cos(\theta_i) \cos(\theta_j)}{2\pi \|c_i - x\|^2} \, dx \]

Form factor from \( j \) to \( i \), \( F_{ij} \)
**Rewrite in Terms of Patches**

\[ H_{x'} = E_{x'} + \rho_{x'} \int_S \delta(x', x'') \frac{H_{x''} \cos(\theta') \cos(\theta'')}{2\pi \|x' - x''\|^2} \, dx'' \]

\[ H_i = E_i + \rho_i \sum_j H_j \int_{S_j} \frac{\cos(\theta_i) \cos(\theta_j)}{2\pi \|c_i - x\|^2} \, dx \]

**Form factor from \( j \) to \( i \), \( F_{ij} \)**

**Example of a rough approximation:**

\[ F_{ij} \approx \delta_{ij} \frac{\cos(\theta_i) \cos(\theta_j)}{2\pi \|c_i - c_j\|^2} A_j \]
Radiosity Method

- Given the $E_i$ and $\rho_i$
- First compute $F_{ij}$
- Then solve
  \[ H_i = E_i + \rho_i \sum_j H_j F_{ij} \]
- Comments:
  - The matrix $A$ is typically very large
  - It is also sparse (why?)
  - Should be solved with an iterative method
    - e.g.: Jacobi or Gauss-Seidel
  - Solution is view independent

\[ h = e + Ah \]
\[ (I - A)h = e \]
Radiosity Method

- Given the light emitted and surface properties
- First compute $F_{ij}$, form factors between patches
- Then **solve a linear system to balance energy between all patches**

Comments:
  - The system is very large
  - It is also sparse (why?)
  - Should be solved with an iterative method
    - e.g.: Jacobi or Gauss-Seidel
  - **Solution is view independent**
Progressive Radiosity

- If magnitude of eigenvalues of $A<1$

\[(I - A)^{-1} = I + A + A^2 + A^3 + \cdots\]

- True for form-factor matrices

- Idea: let important sources of light energy emit first, maybe don’t even bother with dark things

- Use Gauss-Seidel-like iteration but reorder by priority

Southwell Relaxation
Progressive Radiosity

From dissertation "Efficient and predictive realistic image synthesis" by Karol Myszkowski
Each patch will have a constant color

- Smooth solution (e.g. average to vertices)
Other Things

- Each patch will have a constant color
  - Smooth solution (e.g. average to vertices)
- No specular reflection
  - Add Phong specular term or raytraced specular reflection
- Grid artifacts
  - Be clever with grid...
Hierarchical Radiosity

- Light smoothes with distance
  - Compare $1/h^2$ with $1/(h^2 + d^2)$ as $h$ gets large
Hierarchical Radiosity

- **Light smoothes with distance**
  - Compare $1/h^2$ with $1/(h^2 + d^2)$ as $h$ gets large

- **Group patches into hierarchy**
  - Far interactions use lower-res form factors
Computing Form Factors

Form factors have a geometric meaning

Images from SIGGRAPH 93 Education Slide Set by Stephen Spencer
Computing Form Factors

- Form factors have a geometric meaning
- "Hemicube" algorithm uses regular scan conversion

Images from SIGGRAPH 93 Education Slide Set by Stephen Spencer
Computing Form Factors

- Form factors have a geometric meaning.
- “Hemicube” algorithm uses regular scan conversion.
- Also computed by ray-based sampling.
- In practice, computing form factors is the bottleneck.
Photon Mapping

- Lights cast “photons” into environment
  - Cast in random directions
  - Trace into environment
  - Store records at intersections
Photon Mapping

- Lights cast “photons” into environment
  - Cast in random directions
  - Trace into environment
  - Store records at intersections
    - With KD-Trees...
Comparison

Ray Tracing

Ray Tracing w/ Photon Map

Catherine Bendebury and Jonathan Michaels
CS 184 Spring 2005
Photon Mapping

A ray traced image

Note:
Dark shadows
Unlit corners
Nice reflections

Image by Per Christensen
Photon Mapping

Raw photons

Note:
  Noisy
  Sparse

Image by Per Christensen
Photon Mapping

Interpolated Photons

Note:
Still noisy
Biased

Image by Per Christensen
Photon Mapping

Interpolated Photons
(multiplied by diffuse)

Note:
Still noisy
Biased

Image by Per Christensen
Photon Mapping

- **Final Gather**
  - Ray trace scene
  - Direct and specular rays as normal
  - Diffuse rays traced into photon map
  - *Diffuse reflection smoothes noise*
Photon Mapping

Final Image

Note:
Not noisy
Nice lighting
Reflections
May still be biased

Final gather often bottleneck...

Image by Per Christensen
Ambient Occlusion

- A “hack” to create more realistic ambient illumination cheaply
- Assume light from everywhere is partially blocked by local objects
  - At a point on the surface cast rays at random
  - Ambient term is proportional to percent of rays that hit nothing
  - Weight average by cosine of angle with normal
  - Take into account how far before occluded
 Ambient Occlusion

[Diagram showing rays indicating ambient occlusion]
Ambient Occlusion

Diffuse Only  Ambient Occlusion  Combined
Ambient Occlusion

nVidia Gelato Demo Image