Rendering Algorithms:
Radiosity

Spring 2010
Matthias Zwicker
Today

Radiosity

• Theory
• Algorithm
Radiosity

- Finite element method (FEM) to solve rendering equation
  - As opposed to Monte Carlo methods
- Related to heat transfer theory
- Simplifications
  - Assume diffuse reflections on all surfaces
  - Solve for radiant exitance, or radiosity, (power per area) instead of radiance
  - No directional variation, view independent
The rendering equation (revisited)

\[ L(x, \omega_o) = L_c(x, \omega_o) + \int_{H^2(n)} f(x, \omega_o, \omega_i) L(x', -\omega_i) \cos \theta_i d\omega_i \]
Surface form

- Conversion from integral over solid angle to integral over surface area

\[ d\omega_i = \frac{\cos \psi \, dA'}{\|x' - x\|^2} \]

- Geometry term

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

- Visibility term

\[ V(x, x') = \begin{cases} 
1 & \text{if } x \text{ and } x' \text{ are mutually visible} \\
0 & \text{otherwise.}
\end{cases} \]

- Surface form of the rendering equation

\[
L(x, \omega_o) = L_e(x, \omega_o) + \int_S f(x, \omega_o, \omega_i) L(x', -\omega_i) V(x, x') G(x, x') dA'
\]
The rendering equation (revisited)

\[ L(x, \omega_o) = L_e(x, \omega_o) + \int_{H^2(n)} f(x, \omega_o, \omega_i) L(x', -\omega_i) \cos \theta_i d\omega_i \]

\[ L(x, \omega_o) = L_e(x, \omega_o) + \int_S f(x, \omega_o, \omega_i) L(x', -\omega_i) V(x, x') G(x, x') dA' \]
Radiosity equation

- Radiosity, or radiant exitance (power per area) $B(x)$
- Diffuse BRDF $f_d(x) = \frac{\rho(x)}{\pi}$
- Emitted radiosity (light sources) $E(x)$

$$B(x) = E(x) + \int_S f_d(x) B(x') V(x, x') G(x, x') dA'$$

$$= E(x) + \frac{\rho(x)}{\pi} \int_S B(x') V(x, x') G(x, x') dA'$$
Radiosity

“Finite element” approach

- Express solution as a weighted sum of basis functions $\Phi_i(x)$

$$B(x) = \sum_i B_i \Phi_i(x)$$

- Simplest approach: constant basis function on a triangular mesh

$$\Phi_i(x) = \begin{cases} 
1 & x \text{ in triangle } i \\
0 & \text{otherwise}
\end{cases}$$

- Solve for unknown coefficients $B_i$
Finite element approach

- Substitute $B(x) = \sum_i B_i \Phi_i(x)$ into radiosity equation, we get

$$B_i = E_i + \rho_i \sum_j B_j F_{ij}$$

- Form factor $F_{ij}$

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{V(x, x') \cos \theta_i \cos \psi}{\pi \|x' - x\|^2} dA_j dA_i$$

with area of triangles $A_i, A_j$

- Intuition: form factor is “fraction of light leaving triangle $i$ incident on triangle $j$“
Intuition

$\rho_3, E_3$  

$\rho_2, E_2$  

$\rho_1, E_1$
Intuition

\[ \rho_3, E_3 \]

\[ F_{13} \]

\[ \rho_2, E_2 \]

\[ F_{12} \]

\[ \rho_1, E_1 \]

\[ B_1 = E_1 + \rho_1 F_{12} B_2 + \rho_1 F_{13} B_3 \]
Intuition

\[ B_1 = E_1 + \rho_1 F_{12} B_2 + \rho_1 F_{13} B_3 \]

\[ B_2 = E_2 + \rho_2 F_{21} B_1 + \rho_2 F_{23} B_3 \]
**Intuition**

\[ B_1 = E_1 + \rho_1 F_{12} B_2 + \rho_1 F_{13} B_3 \]
\[ B_2 = E_2 + \rho_2 F_{21} B_1 + \rho_2 F_{23} B_3 \]
\[ B_3 = E_3 + \rho_3 F_{31} B_1 + \rho_3 F_{32} B_2 \]
Intuition

\[
\begin{bmatrix}
1 & -\rho_1 F_{12} & -\rho_1 F_{13} \\
-\rho_2 F_{21} & 1 & -\rho_2 F_{23} \\
-\rho_3 F_{31} & -\rho_3 F_{32} & 1
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
B_3
\end{bmatrix}
= \begin{bmatrix}
E_1 \\
E_2 \\
E_3
\end{bmatrix}
\]

„transfer matrix“
Radiosity examples

49 patches per side
constant coloring per patch RGB plot

[Goran et al. 1984]
Matrix formulation

- **Matrix form of** \( B_i = B_{i,e} + \rho_i \sum_j B_j F_{ij} \)

\[
\begin{bmatrix}
1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\
-\rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdots & -\rho_2 F_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
-\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & 1 - \rho_n F_{nn}
\end{bmatrix}
\begin{bmatrix}
B_1 \\
B_2 \\
\vdots \\
B_n
\end{bmatrix}
= \begin{bmatrix}
E_1 \\
E_2 \\
\vdots \\
E_n
\end{bmatrix}
\]

\[
\text{MB} = \text{E}
\]

- **Note that** \( F_{ii} = 0 \) **for planar surface patches (e.g., triangles)**
Today

Radiosity

• Theory

• Algorithm
Radiosity algorithm

1. Mesh Surfaces into Elements
2. Compute Form Factors Between Elements
3. Solve Linear System for Radiosities
4. Reconstruct and Display Solution
Radiosity algorithm

1. Mesh Surfaces into Elements
2. Compute Form Factors Between Elements
3. Solve Linear System for Radiosities
4. Reconstruct and Display Solution
Form factors

- Area-to-area form factor

\[ F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{V(x, x') \cos \theta_i \cos \psi}{\pi \|x' - x\|^2} dA_j dA_i \]

- Area-to-point

\[ F_{dA_i, A_j} = \int_{A_j} \frac{V(x, x') \cos \theta_i \cos \psi}{\pi \|x' - x\|^2} dA_j \]

- Sample area-to-point form factors as an approximation to area-to-area factors
Hemicube Approximation

First radiosity algorithm to deal with occlusion
1. Render scene from the point of view of each vertex/element
2. Compute delta form factors – contribution from each pixel

\[ F_{dA_i,A_j} = \sum_{p \in A_j} \Delta F_p \]

Render source elements from POV of receiving element

Typical resolution: 32x32
Hemicube approximation

\[ r = \sqrt{x^2 + y^2 + 1} \]

\[ \cos \phi = \frac{1}{\sqrt{x^2 + y^2 + 1}} \]

\[ \Delta F = \frac{\Delta A}{\pi (x^2 + y^2 + 1)^2} \]

\[ r = \sqrt{1 + y^2 + z^2} \]

\[ \cos \phi = \frac{1}{\sqrt{1 + y^2 + z^2}} \]

\[ \Delta F = \frac{\Delta A}{\pi (1 + y^2 + z^2)^2} \]
Hemicube approximation

[http://www.siggraph.org/education/materials/HyperGraph/radiosity/overview_2.htm]
Radiosity algorithm

1. **Mesh Surfaces into Elements**
2. **Compute Form Factors Between Elements**
3. **Solve Linear System for Radiosity**
4. **Reconstruct and Display Solution**
Meshing

Reference Solution  Uniform Mesh

Table in room sequence from Cohen and Wallace
Meshing

A. Blocky shadows
B. Missing features
C. Mach bands
D. Inappropriate shading discontinuities
E. Unresolved discontinuities
Meshing

Increasing mesh resolution
Meshing

Adaptive meshing
Meshing

Discontinuity meshing

Discontinuity driven

Regular subdivision
Discontinuity meshing
Solve linear system

\[ MB = E \Rightarrow B = M^{-1}E \]

- **Direct inversion**
  - Gaussian elimination, slow

- **Iterative solvers**
  - Southwell iteration, shooting
Gathering

for (i=0; i<n; i++)
    B[i] = Be[i];

while ( !converged ) {
    for (i=0; i<n; i++) {
        E[i] = 0;
        for (j=0; j<n; j++)
            E[i] += F[i][j]*B[j];
        B[i] = Be[i] + rho[i]*E[i];
    }
}

Row of $F$ times $B$

Calculate one row of $F$ and discard
Gathering

\[ L_e \]
\[ K \circ L_e \]
\[ K \circ K \circ L_e \]
\[ K \circ K \circ K \circ L_e \]

\[ L_e \]
\[ L_e + K \circ L_e \]
\[ L_e + \cdots K^2 \circ L_e \]
\[ L_e + \cdots K^3 \circ L_e \]
Shooting

for (i = 0; i < n; i++) {
    B[i] = dB[i] = Be[i];
    while (!converged) {
        set i st dB[i] is the largest;
        for (j = 0; j < n; j++)
            if (i != j) {
                db = rho[j] * F[j][i] * dB[i];
                dB[j] += db;
                B[j] += db;
            }
        dB[i] = 0;
    }
}

Brightness order

Column of $F$ times $B$
Radiosity examples

[Cohen et al. 1988]
Radiosity examples

[Cohen et al. 1988]
Radiosity summary

• Diffuse reflection only
  - Extensions to non-diffuse surfaces possible, but costly to compute

• Finite element approach
  - Solution is expressed as a sum of basis functions

• Yields linear system of equations
Radiosity

- Advantages over Monte Carlo path tracing
  - No noise
- Major challenges
  - Meshing
  - Form factor computation
  - Solution of linear system
- Often used only for diffuse indirect illumination
  - Other light paths using Monte Carlo path tracing, photon mapping
Radiosity summary

• Recent reincarnation as „precomputed radiance transfer“
  - Precompute (inverse of) transfer matrix
  - Use for real-time shading

• Main limitation: only for static scenes
Radiosity further reading

- “Radiosity and Realistic Image Synthesis”, Cohen, Wallace

- “Advanced Global Illumination”, Dutre, Bekaert, Bala
  http://www.advancedglobalillumination.com/