Radiosity estimates via finite elements

D.A. Forsyth
after slides by John Hart
In a world of diffuse surfaces ...

- Recall
  - radiosity is radiated power per unit area, independent of direction
  - we obtained:

\[
B(x) = E(x) + \rho(x) \int_S \frac{\cos \theta_i \cos \theta_s}{\pi r^2} \text{Vis}(x, u) B(u) dA_s
\]

- which we wrote as:

\[
B(x) - E(x) - \rho(x) \int K(x, u) B(u) dA_u = 0
\]
Radiosity estimate via finite elements

- Divide domain into patches
- Radiosity will be constant on each patch
  - patch basis function, or element
    \[ \phi_i(x) = \begin{cases} 1 & \text{if } x \text{ in patch } i \\ 0 & \text{otherwise} \end{cases} \]
- Now write
  - B_i for radiosity at patch i
  - E_i for exitance at patch i
  - Substitute into eqn:
\[ B(x) - E(x) - \rho(x) \int K(x, u)B(u)dA_u = 0 \]

Becomes

\[ \left( \sum_i B_i \phi_i(x) \right) - \left( \sum_i E_i \phi_i(x) \right) - \left( \rho(x) \int K(x, u) \left( \sum_i B_i(u) \right) dA_u \right) = R(x) \]

This should be “like zero”
Obtaining an estimate: Finite elements

- But in what sense is it zero?
  - Galerkin method

\[
\int R(x) \phi_k(x) dA_x = 0 \forall k
\]

- Apply to:

\[
\left( \sum_i B_i \phi_i(x) \right) - \left( \sum_i E_i \phi_i(x) \right) - \left( \rho(x) \int K(x, u) \left( \sum_i B_i(u) \right) dA_u \right) = R(x)
\]

- And get

\[
B_k A_k - E_k A_k - \sum_j \left( \int_{\text{patch } k} \rho(x) \int_{\text{patch } j} K(x, u) du dx \right) B_j = 0
\]
Finite Element Radiosity Equation

• Start with:

\[ B_k A_k = E_k A_k + \sum_j \left( \int_{\text{patch } k} \rho(x) \int_{\text{patch } j} K(x, u) du dx \right) B_j \]

• Divide through by \( A_k \), assume constant albedo patches, get

\[ B_k = E_k + \sum_j \rho_k F_{jk} B_j \]

• Where geometric effects are concentrated in the form factor

\[ F_{jk} = \frac{1}{A_k} \int_{\text{patch } k} \int_{\text{patch } j} K(x, u) du dx \]
Finite Element Radiosity

- This is a linear system
  \[ B_k = E_k + \sum_j \rho_{kj} F_{jk} B_j \]

- fold in albedo, write
  \[ B_k = E_k + \sum_j \Gamma_{kj} B_j \]

- or in terms of matrices and vectors
  \[ \mathbf{B} = \mathbf{E} + \Gamma \mathbf{B} \]

- **BUT YOU SHOULD NEVER DO:**
  - \( \mathbf{B} = (\mathbf{I} - \Gamma)^{-1} \mathbf{E} \)
  - \( \mathbf{B} \) might have \( 10^6 \) elements or more!
Form factors

• if patches are all flat, then: \( F_{ii} = 0 \)

• if i can’t see j at all, then: \( F_{ij} = 0 \)

• reciprocity: \( A_k F_{jk} = A_j F_{kj} \)

• interpretation:
  • \( F_{jk} \) is percentage of energy leaving \( k \) that arrives at \( j \)
  • this gives:

\[
\sum_j F_{jk} = 1
\]
Computing form factors

- Nusselt’s analogy

\[ F_{ij} = \frac{\text{proj}_D(\text{proj}_\Omega(A_j))}{\text{Area}(D)} \]
The Hemicube

- Render onto faces of cube on receiver

\[ \Delta F_{dAiA_j} = \frac{\cos\phi_i \cos\phi_j}{\pi r^2} \Delta A \]
Random samples

- with uniform samples

\[ A_k F_{jk} = \frac{1}{N} \sum \frac{\cos \theta_i \cos \theta_j \text{Vis}(i, j)}{\pi r^2} \]
Solving the radiosity system: Gathering

- Neumann series (again!) \[ B = E + \Gamma E + \Gamma^2 E + \Gamma^3 E + \ldots \]

- Easy iteration

\[ B^{(0)} = E \]
\[ B^{(n+1)} = E + \Gamma B^{(n)} \]
Gathering with iterative methods

- Linear system \( Ax = b \)

\[ \sum_j a_{ij} x_j = b_i \]

- Jacobi iteration
  - reestimate each \( x \)

\[ x_j^{(n+1)} = \frac{1}{a_{jj}} \left( b_i - \sum_{l \neq j} a_{il} x_l^{(n)} \right) \]

- Gauss-Seidel
  - reuse new estimates

\[ x_j^{(n+1)} = \frac{1}{a_{jj}} \left( b_i - \sum_{l < j} a_{il} x_l^{(n+1)} - \sum_{l > j} a_{il} x_l^{(n)} \right) \]
From Cohen, SIGGRAPH 88
Southwell iteration: Progressive radiosity

• Gauss-Seidel, Jacobi, Neumann require us to evaluate whole kernel at each iteration
  • this is vilely expensive 10^6 x 10^6 matrix?
  • it’s also irrational
    • in G-S, Jacobi, for one pass through the variables,
      • we gather at each patch, from each patch
        • but some patches are not significant sources
    • we should like to gather only from bright patches
      • or rather, patches should “shoot”

• This is Southwell iteration