Radiosity

Thomas A. Funkhouser

Overview

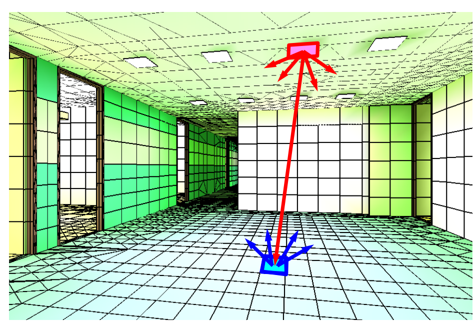
- **Rendering equation**
- **Radiosity equation**
- **Radiosity methods**
  - Computing form factors
  - Solving the linear system
  - Meshing adaptively
- **Comparison to ray tracing**

Radiosity

- **Goals**
  - Simulate diffuse inter−object reflections and shadows

Radiosity

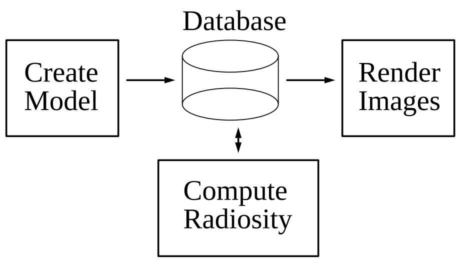
- **Basic idea**
  - Treat every polygon as light source

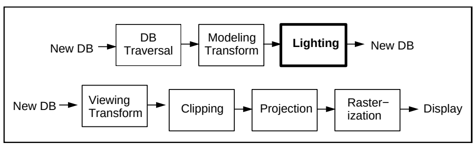


- **Advantages**
  - Physically models shadows and indirect diffuse illumination
  - Independent of any viewpoint

Radiosity Pipeline

- **Two−Phase Rendering:**





Radiosity Equation

- **Equation Formulation:**

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

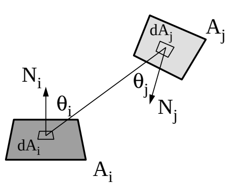
B_i = Radiosity of patch i
E_i = Emission of patch i
p_i = Reflectivity of patch i (K_D)
F_{ij} = Form−factor between patches i and j

- **Matrix Formulation:**

$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \bullet & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \bullet & -\rho_2 F_{2n} \\ -\rho_3 F_{31} & -\rho_3 F_{32} & \bullet & -\rho_3 F_{3n} \\ \bullet & \bullet & \bullet & \bullet \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \bullet & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ \bullet \\ E_n \end{bmatrix}$$

Form Factors

- **Definition:**
  - − Fraction of energy leaving patch j that arrives at patch i



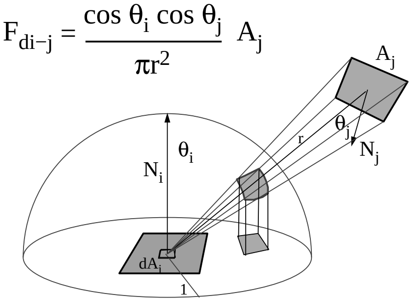
$$F_{di-dj} = \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij}\, dA_j$$

$$F_{ij} = \frac{1}{A_i} \int_{A_i}\int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij}\, dA_j dA_i$$

Computing Form Factors

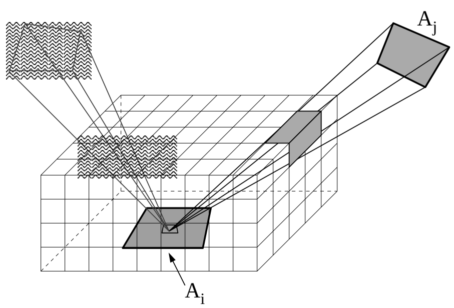
- **Method:**
  - − Project onto unit hemisphere ($A_j \cos\theta_j / r^2$)
  - − Project onto unit circle base ($\cos\theta_i$)
  - − Divide by area of circle ($1/\pi$)

$$F_{di-j} = \frac{\cos\theta_i \cos\theta_j}{\pi r^2} A_j$$



Hemi−Cube Form Factors

- **Method:**
  - Project scene onto hemi−cube positioned at centroid of patch i
  - Count pixel coverage to determine form factors F_{i*}
  - Can use hardware z−buffer
  - Image precision (aliasing)



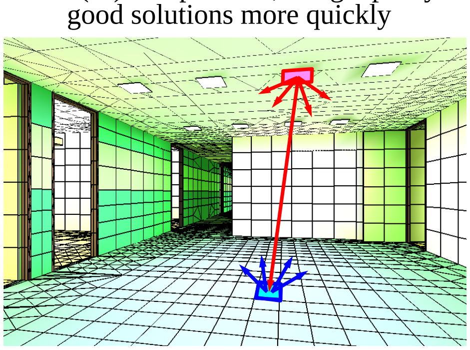
Matrix Solution Methods

- **Invert the Matrix:**
  - Gaussian elimination
  - $O(n^3)$
- **Gauss−Seidel Iteration**
  - Iterate gather radiosity to patches
  - Relax rows of variables
  - $O(n^2)$

$$\begin{bmatrix} 1-\rho_1F_{11} & -\rho_1F_{12} & \cdot & -\rho_1F_{1n} \\ -\rho_2F_{21} & 1-\rho_2F_{22} & \cdot & -\rho_2F_{2n} \\ -\rho_3F_{31} & -\rho_3F_{32} & \cdot & -\rho_3F_{3n} \\ \cdot & \cdot & \cdot & \cdot \\ -\rho_nF_{n1} & -\rho_nF_{n2} & \cdot & 1-\rho_nF_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ E_3 \\ \cdot \\ E_n \end{bmatrix}$$

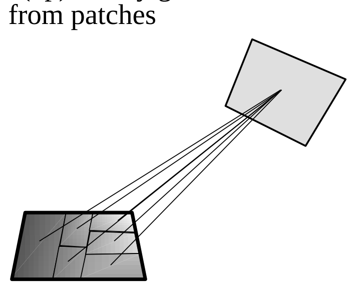
Matrix Solution Methods

- **Progressive refinement**
  - Iterate shoot radiosity from patches
  - Relax columns of variables
  - Select patches to shoot in order of unshot radiosity
  - $O(n^2)$ computation, but get pretty good solutions more quickly



Mesh Substructuring

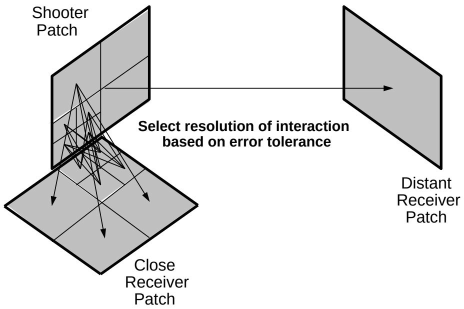
- **Adaptive subdivision**
  - Partition each patch into multiple ‘‘elements’’ (along gradients of radiosity)
  - O(np) if only gather to elements from patches



Patch partitioned into 6 elements

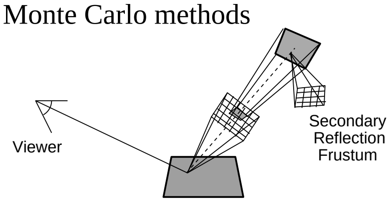
Hierarchical Radiosity

- **Multiresolution Computation**
  - Substructure patches into quad−tree
  - Transfer energy using lower resolution mesh elements if can do so within error tolerance
  - O(n) computation



Ray Tracing and Radiosity

- **Dilemma:**
  - Radiosity is good at diffuse interobject reflection
  - Ray tracing is good at specular interobject reflection
- **Combine them**
  - Example: compute diffuse interobject reflections in a ray tracer
  - Monte Carlo methods



Summary

- **Radiosity equation**
- **Computing form factors**
- **Solution methods**
- **Combining radiosity and ray tracing**