Radiosity

Tom Funkhouser

COS 526, Fall 2012

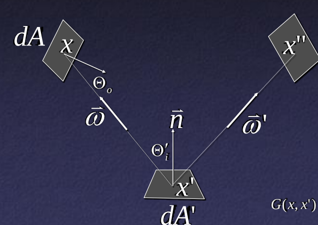
Radiosity

Overview

- Radiosity equation
- Solution methods
  - Computing form factors
  - Selecting basis functions for radiosities
  - Solving linear system of equations
  - Meshing surfaces into elements
  - Rendering images

Rendering Equation

$$L(x' \to x'') = L_e(x' \to x'') + \int_S f_r(x \to x' \to x'') L(x \to x') V(x,x') G(x,x') dA$$



$$G(x,x') = \frac{\cos\Theta_i' \cos\Theta_o}{\|x - x'\|^2}$$

Radiosity Equation

$$L(x' \to x'') = L_e(x' \to x'') + \int_S f_r(x \to x' \to x'') L(x \to x') V(x,x') G(x,x') dA$$

Assume everything is Lambertian

$$f_r(x \to x' \to x'') = \rho(x')/\pi$$

$$L(x') = L_e(x') + \frac{\rho(x')}{\pi} \int_S L(x) V(x,x') G(x,x') dA$$

Convert to Radiosities

$$B = \int_\Omega L_o \cos\theta \, d\omega \qquad B = \pi L$$

$$B(x') = B_e(x') + \rho(x') \int_S B(x) V(x,x') G(x,x') dA$$

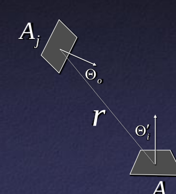
Radiosity Approximation

$$B(x') = B_e(x') + \rho(x') \int_S B(x) V(x,x') G(x,x') dA$$

Discretize surfaces into elements

$$B_i = E_i + \rho_i \sum_{j=1}^{N} B_j F_{ij}$$

where $F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{V_{ij} \cos\Theta_i' \cos\Theta_o}{\pi r^2} dA_j dA_i$



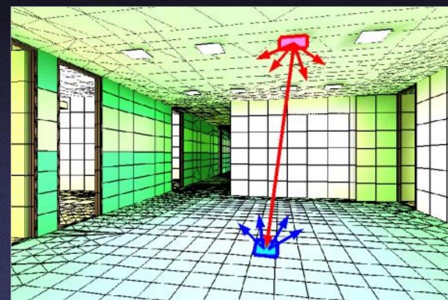
System of Equations

$$B_i = E_i + \rho_i \sum_{j=1}^{N} B_j F_{ij}$$

$$B_i - \rho_i \sum_{j=1}^{N} B_j F_{ij} = E_i$$

$$\begin{bmatrix} 1-\rho_1 F_{1,1} & . & . & . & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & 1-\rho_2 F_{2,2} & . & . & -\rho_2 F_{2,n} \\ . & . & . & . & . \\ . & . & . & . & . \\ -\rho_{n-1} F_{n-1,1} & . & . & . & -\rho_{n-1} F_{n-1,n} \\ -\rho_n F_{n,1} & . & . & . & 1-\rho_n F_{n,n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ . \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ . \\ . \\ . \\ E_n \end{bmatrix}$$

Intuition



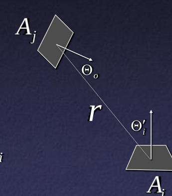
Overview

- Radiosity equation
- Solution methods
  - Computing form factors
  - Selecting basis functions for radiosities
  - Solving linear system of equations
  - Meshing surfaces into elements
  - Rendering images

Form Factor

- Fraction of energy leaving element i that arrives at element j

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{V_{ij} \cos\Theta_i' \cos\Theta_o}{\pi r^2} dA_j dA_i$$



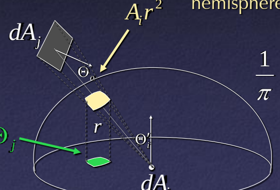
Form Factor Intuition

$$F_{di-dj} = \frac{1}{A_i} \frac{V_{ij} \cos\Theta_i \cos\Theta_j}{\pi r^2}$$

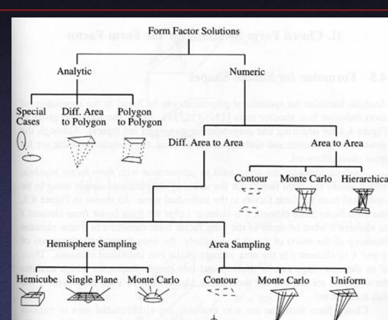
$\frac{\cos\Theta_i}{A_i r^2}$ Projection to hemisphere

Projection to disk $\cos\Theta_j$

$\frac{1}{\pi}$ Divide by area of disk

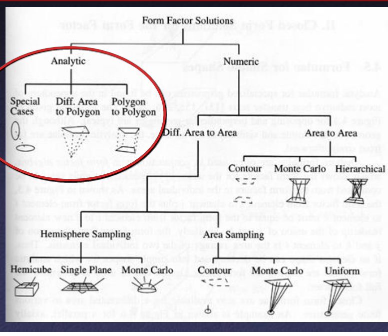


Computing Form Factors



Cohen & Wallace

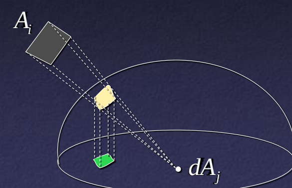
Computing Form Factors



Cohen & Wallace

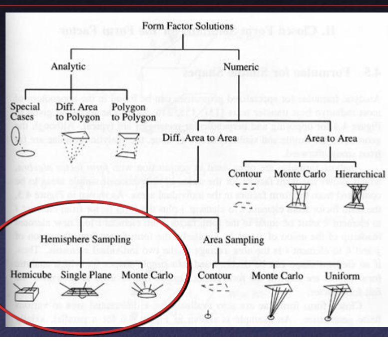
Analytic Form Factors

- Derive equation for projected area
  - Possible only for simple cases



Partial visibility is problematic

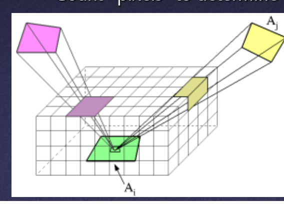
Computing Form Factors



Cohen & Wallace

Hemicube

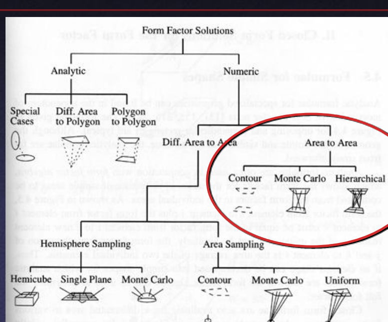
- Compute form factor with image-space precision
  - Render scene from centroid of A_i
  - Use z-buffer to determine visibility of other surfaces
  - Count "pixels" to determine projected areas



Approximating A_i with point leads to errors

Regular sampling leads to aliasing artifacts

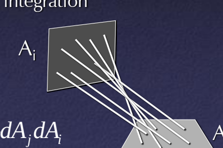
Computing Form Factors



Cohen & Wallace

Monte Carlo Sampling

- Compute form factor by random sampling
  - Select random points on elements
  - Intersect line segment to evaluate V_{ij}
  - Evaluate F_{ij} by Monte Carlo integration



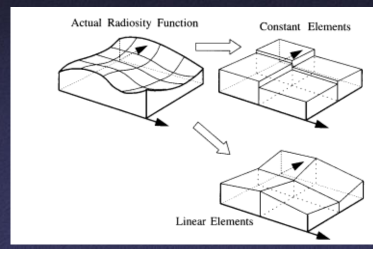
$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{V_{ij} \cos \Theta_i' \cos \Theta_o}{\pi r^2} dA_j dA_i$$

Overview

- Radiosity equation
- Solution methods
  - Computing form factors
  - Selecting basis functions for radiosities
  - Solving linear system of equations
  - Meshing surfaces into elements
  - Rendering images

Selecting a Basis Function

- Store radiosity function on surface mesh
  - Piecewise-constant, piecewise-linear, wavelets, etc.



Cohen & Wallace

Overview

- Radiosity equation
- Solution methods
  - Computing form factors
  - Selecting basis functions for radiosities
  - Solving linear system of equations
  - Meshing surfaces into elements
  - Rendering images

Solving the System of Equations

- Challenges:
  - Size of matrix
  - Cost of computing form factors
  - Computational complexity

$$\begin{bmatrix} 1-\rho_1 F_{1,1} & . & . & . & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & 1-\rho_2 F_{2,2} & . & . & -\rho_2 F_{2,n} \\ . & . & . & . & . \\ . & . & . & . & . \\ -\rho_{n-1} F_{n-1,1} & . & . & . & -\rho_{n-1} F_{n-1,n} \\ -\rho_n F_{n,1} & . & . & . & 1-\rho_n F_{n,n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ . \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ . \\ . \\ . \\ E_n \end{bmatrix}$$

Solving the System of Equations

- Solution methods:
  - Invert the matrix – $O(n^3)$
  - Iterative methods – $O(n^2)$
  - Hierarchical methods – $O(n)$

Gauss-Seidel Iteration

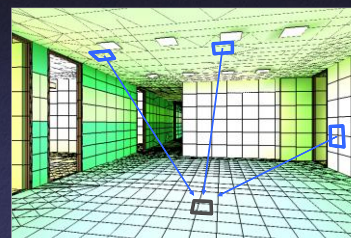
```
for all i
    B_i = E_i
while not converged
    for each i in turn
        B_i = E_i + ρ_i Σ_{j≠i} B_j F_ij
display the image using B_i as the intensity of patch i.
```

Gauss-Seidel Iteration

- Iteratively relax rows of linear system
- Effectiveness depends on sparsity of matrix

Gauss-Seidel Iteration

- Interpretation: gather radiosity to elements

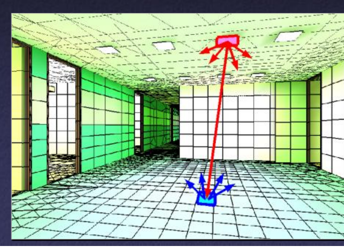


Progressive Radiosity

```
for all i
    B_i = E_i
    ΔB_i = E_i
while not converged
    pick i, such that ΔB_i * A_i is largest
    for every patch j
        Δrad = ΔB_i * ρ_j F_ji
        ΔB_j = ΔB_j + Δrad
        B_j = B_j + Δrad
    ΔB_i = 0
    display the image using B_i as the intensity of patch i.
```

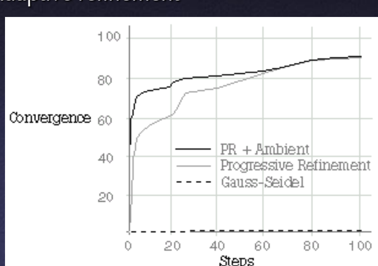
Progressive Radiosity

- Iteratively shoot "unshot" radiosity from elements
- Select shooters in order of unshot radiosity



Progressive Radiosity

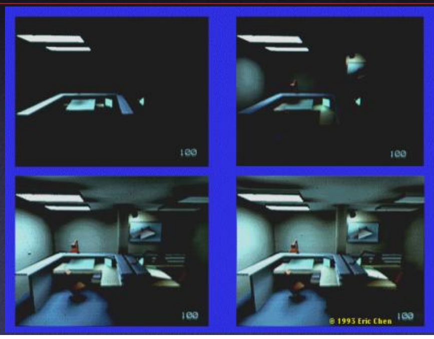
- Adaptive refinement



Yeap

Progressive Radiosity

Progressive Radiosity



Overview

- Radiosity equation
- Solution methods
  - Computing form factors
  - Selecting basis functions for radiosities
  - Solving linear system of equations
  - Meshing surfaces into elements
  - Rendering images

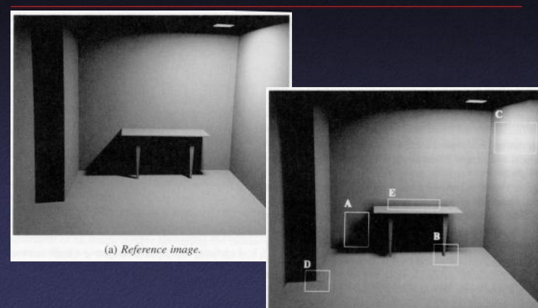
Surface Meshing Goals

- Store radiosity across surface
  - Represents function well
  - Few elements
  - Few visible artifacts

Figure 6.2: *A radiosity image computed using a uniform mesh.*

Cohen & Wallace

Artifacts of Bad Surface Meshing



(a) *Reference image.*

(b) *Artifacts introduced by the approximation.*

Cohen & Wallace

Error Image

Figure 6.4: *Error image.*

(a) *Reference image.*

(b) *Artifacts introduced by the approximation.*

Cohen & Wallace

Error Image

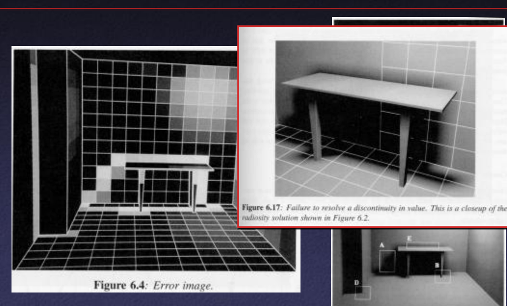


Figure 6.4: *Error image.*

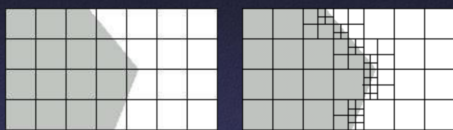
Figure 6.17: *Failure to resolve a discontinuity in value. This is a closeup of the radiosity solution shown in Figure 6.2.*

(b) *Artifacts introduced by the approximation.*

Cohen & Wallace

Adaptive Meshing

- Refine mesh in areas of high residual



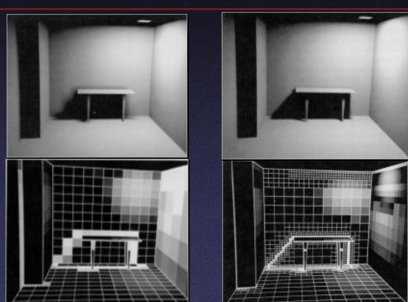
Yeap

Adaptive Meshing

Uniform mesh | Adaptive mesh

Cohen & Wallace

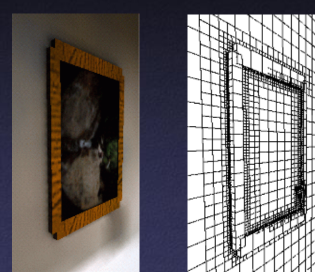
Error Comparison



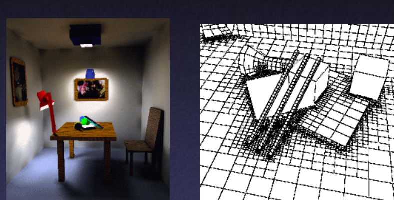
Uniform | Adaptive

Cohen & Wallace

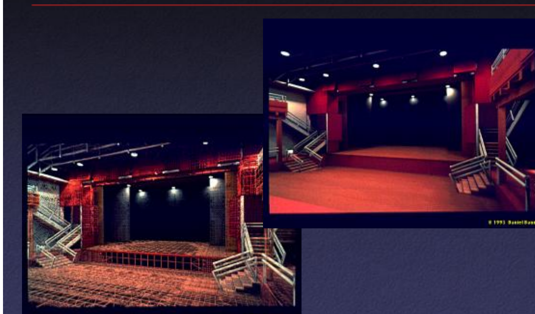
Adaptive Meshing



Adaptive Meshing



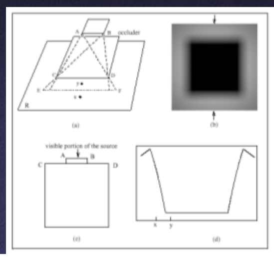
Adaptive Meshing



Baum

Discontinuity Meshing

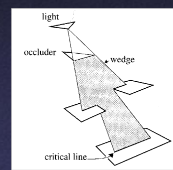
- Capture discontinuities in radiosity across a surface with explicit mesh boundaries



Lischinski

Discontinuity Meshing

- Capture discontinuities in radiosity across a surface with explicit mesh boundaries

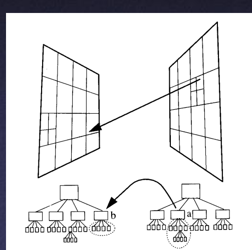


Discontinuity Mesh

Lischinski

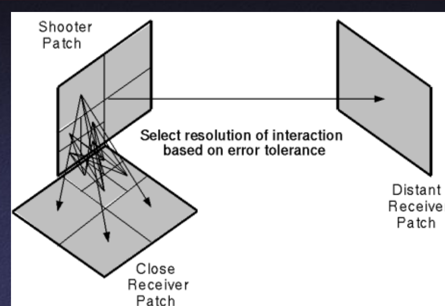
Hierarchical Radiosity

- Estimate errors, refine elements if too large



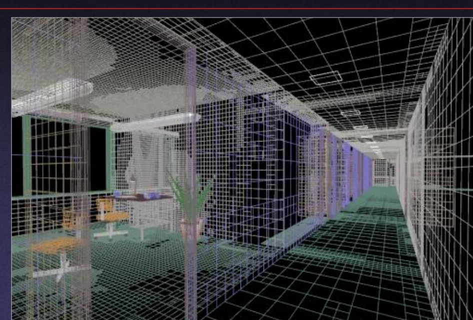
Cohen & Wallace

Hierarchical Radiosity



Hierarchical Radiosity

Hierarchical Radiosity



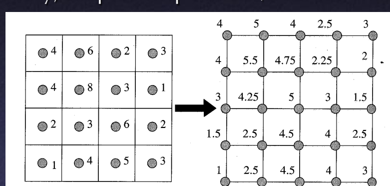
Hierarchical Radiosity

Overview

- Radiosity equation
- Solution methods
  - Computing form factors
  - Selecting basis functions for radiosities
  - Solving linear system of equations
  - Meshing surfaces into elements
  - Rendering images

Displaying Radiosity

- Usually, simple interpolation (Gouraud shading)



computed displayed

- Can also try to preserve discontinuities…

Cohen & Wallace

Extensions

- Non-diffuse environments
  - Directional radiosity functions
  - Extended form factors
  - Multipass methods
- Participating media
  - Path integrals in form factors
- Dynamic scenes
  - Incremental updates
- Parallel solvers
  - Decomposition
  - Scheduling