Practical Solutions

Two approaches

- Monte Carlo methods
- Finite element methods

Classic radiosity

- Diffuse, polygonal surfaces
  - View independent solution
  - Polygonal mesh
- Form factors
- Solving linear equations

Examples

Goral

Nishita, Computer room

Cohen, Vermeer

Cohen, Museum

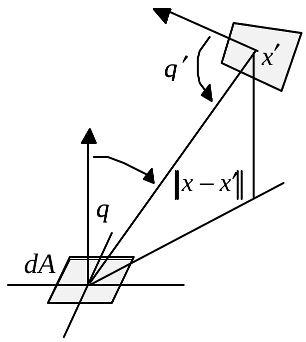
Wallace, Engine room

Lightscape

The Radiosity Equation

Assume diffuse reflection only

Solve for radiosity (2d function)

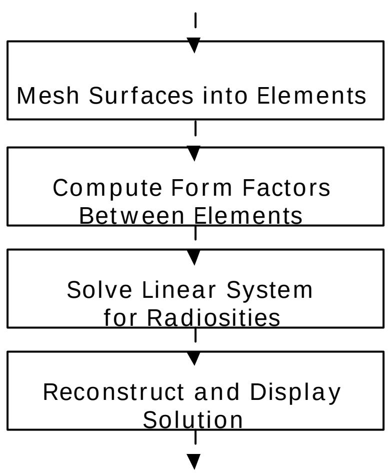


$$B(x) = B_e(x) + \rho(x) E(x)$$

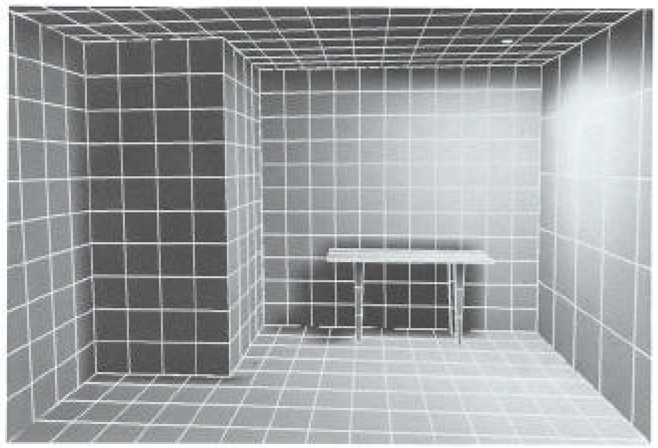
$$B(x) = B_e(x) + \rho(x) \int_{M^2} F(x,x') B(x')\, dA'$$

$$F(x,x') = \frac{G(x,x')}{\pi} = \frac{\cos\theta \cos\theta'}{\pi \|x - x'\|^2} V(x,x')$$

Classic Radiosity Algorithm



Simple Room Scene



Example from John Wallace

Derivation

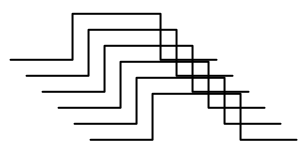
Radiosity integral equation

$$B(x) = B_e(x) + \rho(x) \int_{M^2} B(x')F(x,x')\,dA'$$

Piecewise constant basis functions

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

$$B_e(x) = \sum_i E_i N_i(x)$$



$$\sum B_i N_i(x) = \sum E_i N_i(x) + \rho_i \int_{M^2} F(x,x') \sum B_j N_j(x')\,dA'$$

Derivation

Radiosity integral equation

$$\sum B_i N_i(x) = \sum E_i N_i(x) + r_i \int_{M^2} F(x,x') \sum B_j N_j(x')\, dA'$$

$$\int \left(\sum B_i N_i(x) = \sum E_i N_i(x) + r_i \int_{M^2} F(x,x') \sum B_j N_j(x')\, dA' \right) N_j(x)\, dA$$

$$B_i A_i = E_i A_i + r_i \sum_j B_j \int_{M^2} \int_{M^2} F(x,x') N_i(x) N_j(x')\, dA\, dA'$$

Form Factor

Throughput

$$T_{ij} = T_{ji} = \int_{A_i} \int_{A_j} \frac{\cos q_o' \cos q_i}{p \left\| x - x' \right\|^2} V(x,x')\, dA'\, dA$$

Reciprocity

$$T_{ij} = A_i F_{ij}$$

$$T_{ji} = A_j F_{ji}$$

$$A_i F_{ij} = A_j F_{ji}$$

Classic Radiosity

Power Balance

$$B_i A_i = E_i A_i + r_i \sum_j B_j A_j F_{ji}$$

Reciprocity

$$A_i F_{ij} = A_j F_{ji} \Rightarrow B_i = E_i + r_i \sum_j F_{ij} B_j$$

Radiosity System

$$\begin{pmatrix} 1 - r_1 F_{11} & -r_1 F_{12} & \cdots & -r_1 F_{1n} \\ -r_2 F_{21} & 1 - r_2 F_{22} & \cdots & -r_2 F_{21} \\ \vdots & \vdots & \ddots & \vdots \\ -r_n F_{n1} & -r_n F_{n2} & \cdots & 1 - r_n F_{nn} \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{pmatrix} = \begin{pmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{pmatrix}$$

Form Factor Properties

Summation

$$\sum_j F_{ij} = \sum_i F_{ji} = 1$$

Form factor is the percentage of light...

Form Factors

Differential-differential

$$F_{dA_i,dA_j} = \frac{\cos\theta_o' \cos\theta_i}{\pi \left\| x - x' \right\|^2} V(x,x')\, dA_j$$

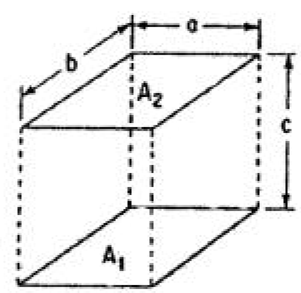
Differential-finite

$$F_{dA_i,A_j} = \int_{A_j} \frac{\cos\theta_o' \cos\theta_i}{\pi \left\| x - x' \right\|^2} V(x,x')\, dA'$$

Finite-finite

$$F_{A_i,A_j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_o' \cos\theta_i}{\pi \left\| x - x' \right\|^2} V(x,x')\, dA'\, dA$$

Analytical Form Factors



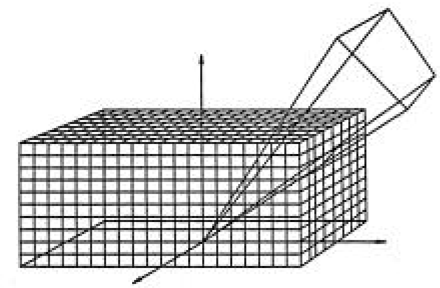
$$X = \frac{a}{c}$$

$$Y = \frac{b}{c}$$

$$F_{A_1,A_2} = \frac{2}{\pi XY} \left\{ \begin{array}{l} \ln\left[\frac{(1+X^2)(1+Y^2)}{(1+X^2+Y^2)} \right]^{1/2} + X\sqrt{1+Y^2}\tan^{-1}\frac{X}{\sqrt{1+Y^2}} \\ +Y\sqrt{1+X^2}\tan^{-1}\frac{Y}{\sqrt{1+X^2}} - X\tan^{-1}X - Y\tan^{-1}Y \end{array} \right\}$$

Hemicube Algorithm

First radiosity algorithm to deal with occlusion



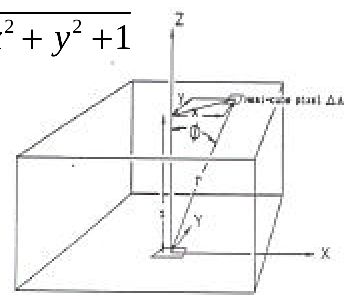
Render source elements from POV of receiving element

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

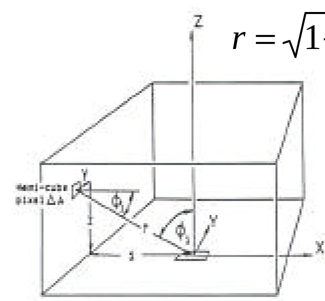
Typical resolution: 32x32

Hemicube Delta Form Factors

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



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



$$\cos f = \frac{1}{\sqrt{x^2 + y^2 + 1}}$$

$$\cos f = \frac{1}{\sqrt{1 + y^2 + z^2}}$$

$$\Delta F = \frac{\Delta A}{p\,(x^2 + y^2 + 1)^2}$$

$$\Delta F = \frac{\Delta A}{p\,(1 + y^2 + z^2)^2}$$

Hemicube Algorithms

Advantages

+ First practical method -> Patent!
+ Use existing rendering systems; Hardware!
+ Computes all form factors in $O(n)$

Disadvantages

- Computes differential-finite form factor
- Aliasing errors due to sampling
  Randomly rotate/shear hemicube
- Proximity errors
- Visibility errors
- Expensive to compute a single form factor

Solve [F][B] = [E]

Direct methods: $O(n^3)$

- Gaussian elimination
  Goral, Torrance, Greenberg, Battaile, 1984

Iterative methods: $O(n^2)$

Convergence

Energy conservation -> diagonally dominant -> converge

- Gauss-Seidel, Jacobi: <u>Gathering</u>
  Nishita, Nakamae, 1985
  Cohen, Greenberg, 1985
- Southwell: <u>Shooting</u>
  Cohen, Chen, Wallace, Greenberg, 1988

Iterative Solvers

Iteration

$$(I-F)^{-1}B = E$$
$$B = (I + B + B^2 + \cdots)E$$

$$B^0 = E$$
$$B^1 = E + FB^0$$
$$\dots$$
$$B^n = E + FB^{n-1}$$

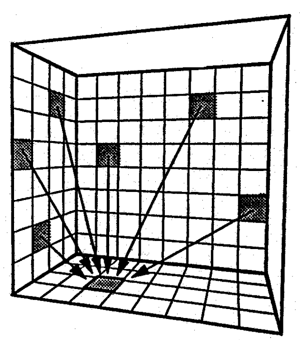
Relaxation

Residual $r^n = E - (I-F)B^n$

Iteration $r_i^{k+1} = 0 \Rightarrow B_i^{k+1} = B_i^k + r_i^k = E_i + r_i \sum_{i \neq j} F_{ij} B_j^k$

If residual is 0, solution has been reached

Gathering

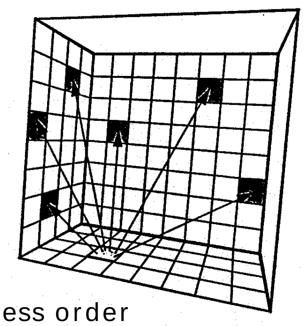


```
for(i=0; i<n; i++)
  B[i] = Be[i];

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] = rho[i]*E[i];
  }
}
```

Scan through elements in "model" order
Row of F times B
Successively set residual to 0
May update radiosities at end (Jacobi), or during (GS)
Calculate one row of F and discard

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[j] = rho[j]*F[j][i]*dB[i];
      dB[j] += dBj;
      B[j] += dBj;
    }
  dB[i]=0;
}
```

Brightness order

Column of F times B

In terms of residuals

- Choose element with maximum residual
- Relax such that that elements residual is 0
- Incrementally update other residuals

Results: Gathering vs. Shooting

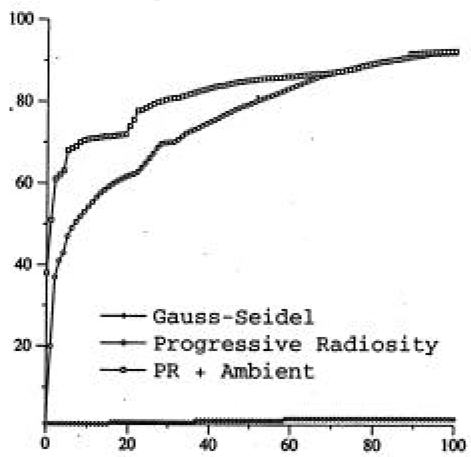


Figure 5.9: *Convergence versus number of steps for three algorithms.*

From Cohen et al.