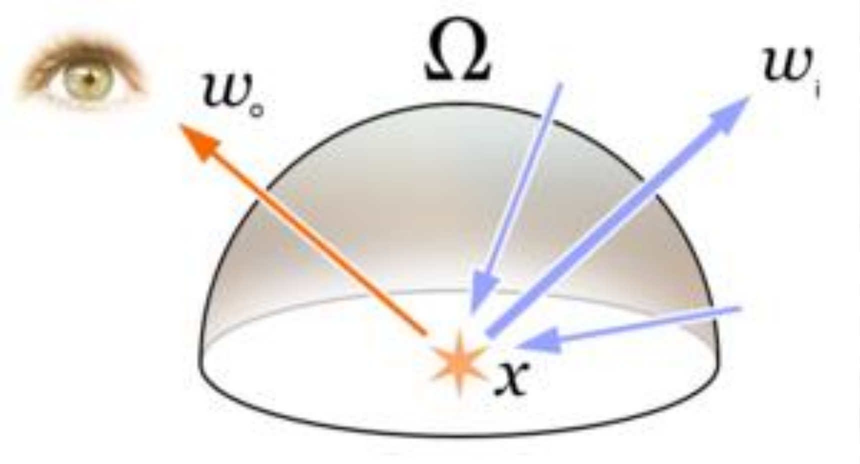
Rendering Equation

$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_\Omega f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t)\, L_i(\mathbf{x}, \omega_i, \lambda, t)\, (\omega_i \cdot \mathbf{n})\, d\omega_i$$

- Linear equation
- Spatial homogeneous
- Both ray tracing and radiosity can be considered special case of this general eq.



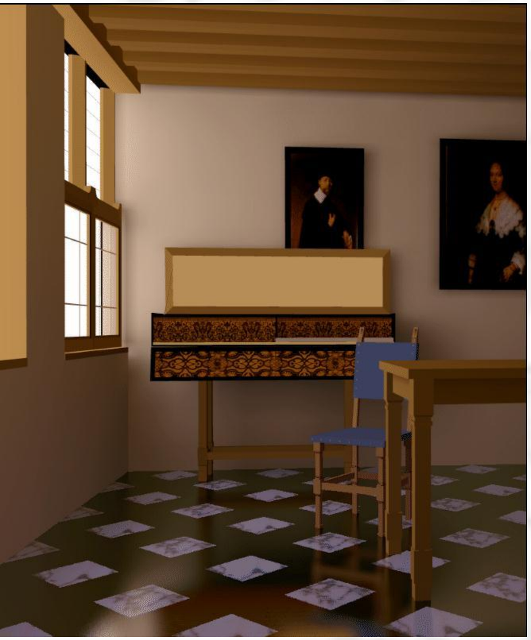
Radiosity

Reality (actual photograph)…

Minus **Radiosity** Rendering…

Equals the **difference** (or error) image

http://www.graphics.cornell.edu/online/box/compare.html



University of California Santa Barbara

Comparison

Ray tracing	Radiosity
View point dependent	View point independent
Specular	Diffuse

Radiosity

- Thermal heat transfer
  - Light transport: transfer of energy from thermally excited surface
- Radiosity: rate at which energe leaves a surface
  - Emitted + reflected
  - Balance (equilibrium) determine the balance of incoming and outgoing flux

Radiosity

- the amount of light (energy) that leaves a surface, including
  - self-emitting energy (source)
  - reflected and/or transmitted energy
  - radiosity = emission + bi-directional reflection
  - bi-directional reflection counts both reflection and transmission transport of both specular and diffuse components
  - bi-directional reflection is a function of
    - radiosities of all other objects in the environment
    - how much received by the particular object

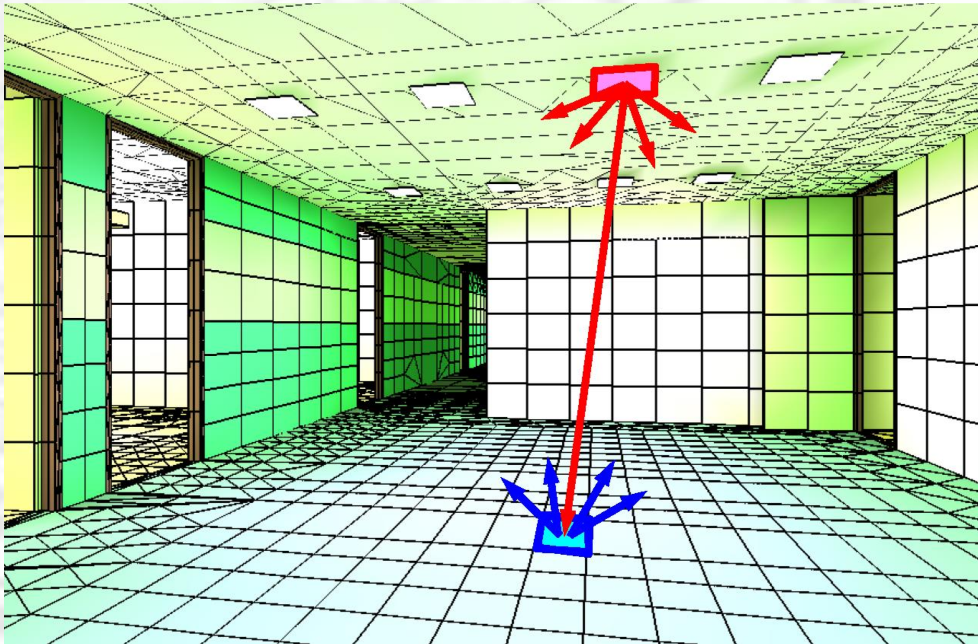
Mathematically

$$radiosity_i = emission_i + reflectivity_i \sum_j radiosity_j \, form\_factor_{j \to i}$$

$$B_i A_i = E_i A_i + \rho_i \sum_j B_j A_j F_{j \to i}$$

$$B_i, E_i : energy/(area \cdot time)$$

$$F_{j \to i} : \frac{total\ energy\ received\ at\ patch\ i\ from\ path\ j}{total\ energy\ from\ patch\ j}$$



$$B_i A_i = E_i A_i + \rho_i \sum_j B_j A_j F_{j\to i}$$

$$B_i = E_i + \rho_i \sum_j B_j F_{j\to i} \frac{A_j}{A_i}$$

$$B_i = E_i + \rho_i \sum_j B_j F_{i\to j} \qquad (F_{j\to i} A_j = F_{i\to j} A_i)$$

$$B_i - \rho_i \sum_j B_j F_{i\to j} = E_i$$

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

Implementation Details

- Reflectivity and emission may be functions of wavelength, hence, the equation may represent a family of equations (e.g., for red, green, and blue channels)
- the form factors depend only on geometry

Implementation Details

- In general
  - the matrix can be very big (e.g., with 1000 patches the matrix is 1000x1000 or with one million entries)
  - it is usually not sparse (nor tri-diagonal, nor banded limited, etc. etc.)
  - iterative solution (e.g., Gauss-Seidal)

- Initially, all B's can be approximated by E's
- Order patches with sources first
- Patches adjacent to sources lit up, then they light up other patches …
- Iterate until the numbers stablize

$$a_{11}B_1^{(n)} = -(a_{12}B_2^{(n-1)} + a_{13}B_3^{(n-1)} + \cdots + a_{1n}B_n^{(n-1)}) + E_1$$

$$a_{22}B_2^{(n)} = -(a_{21}B_1^{(n)} + a_{23}B_3^{(n-1)} + \cdots + a_{2n}B_n^{(n-1)}) + E_2$$

…

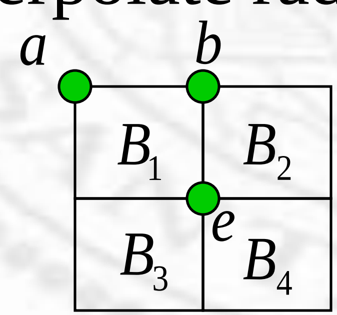
$$a_{ii}B_i^{(n)} = -(a_{i1}B_1^{(n)} + a_{i2}B_2^{(n)} + \cdots + a_{i,i-1}B_{i-1}^{(n)} + a_{i,i+1}B_{i+1}^{(n-1)} + \cdots + a_{in}B_n^{(n-1)}) + E_i$$

…

$$a_{nn}B_n^{(n)} = -(a_{n1}B_1^{(n)} + a_{n2}B_2^{(n)} + \cdots + a_{n,n-1}B_{n-1}^{(n)}) + E_n$$

Standard radiosity methods

- Compute the form factors
- Solve the radiosity matrix equation using Gauss-Seidal method
- Rendering
  - select viewing direction
  - determine visible surfaces
  - interpolate radiosity values



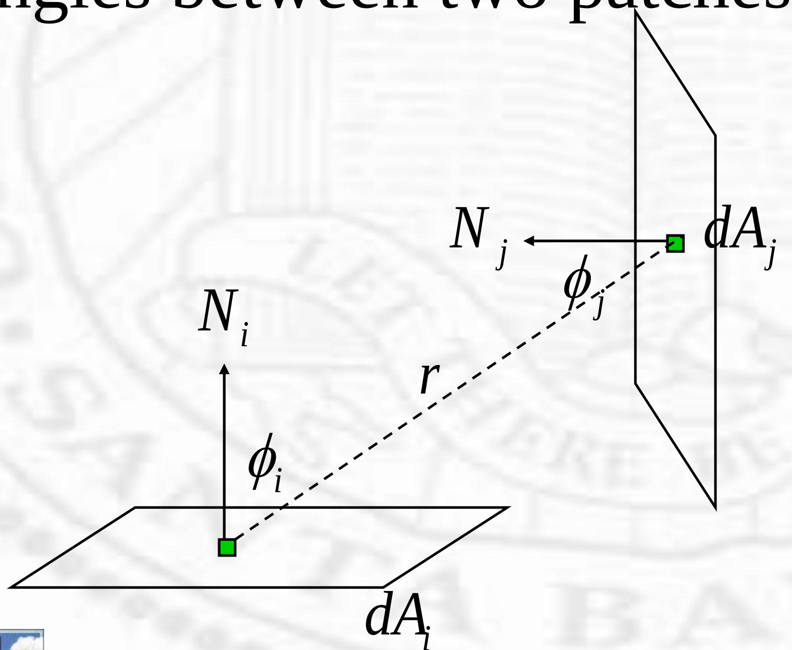
$$B_e = \frac{B_1 + B_2 + B_3 + B_4}{4}$$

$$B_b + B_e = B_1 + B_2 \qquad B_b = 3\frac{B_1 + B_2}{4} - \frac{B_3 + B_4}{4}$$

$$\frac{B_a + B_e}{2} = B_1 \qquad B_a = 7\frac{B_1}{4} - \frac{B_2 + B_3 + B_4}{4}$$

Form Factors

- Without being mathematically rigorous, form factors are affected by
  - distance between two patches
  - angles between two patches



$$F_{dA_i \rightarrow dA_j} = H_{ij} \frac{\cos \phi_i \cos \phi_j}{\pi r^2}$$

$$H_{ij} \begin{cases} 1 & visible \\ 0 & not \end{cases}$$

$$F_{dA_i \to dA_j} = H_{ij} \frac{\cos\phi_i \cos\phi_j}{\pi r^2}$$

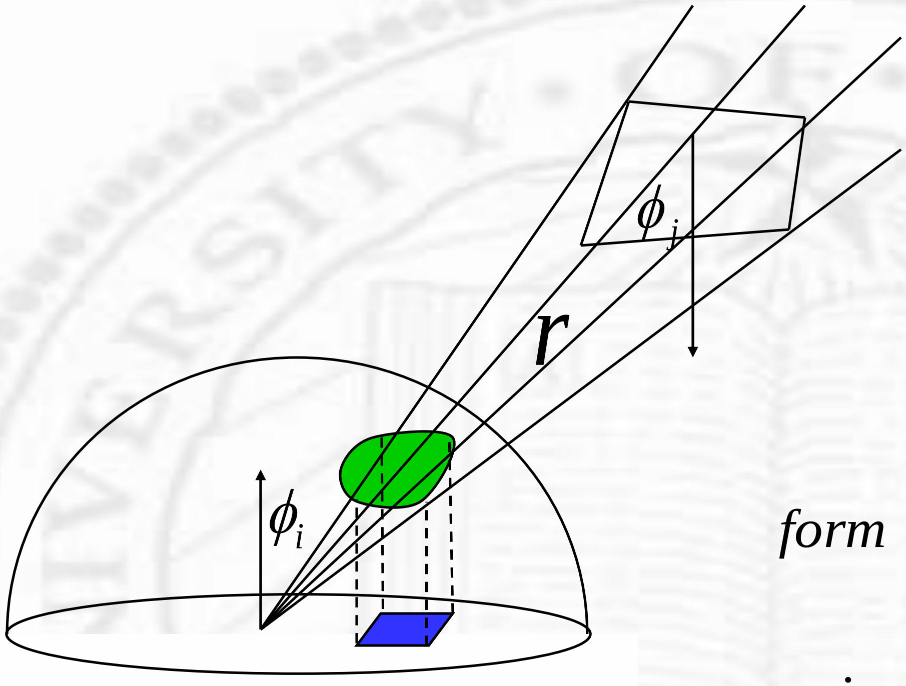
$$F_{dA_i \to A_j} = \int_{A_j} F_{dA_i \to dA_j} dA_j = \int_{A_j} H_{ij} \frac{\cos\phi_i \cos\phi_j}{\pi r^2} dA_j$$

$$F_{A_i \to A_j} = \frac{1}{A_i} \int_{A_i} F_{dA_i \to A_j} dA_i = \frac{1}{A_i} \int_{A_i} \int_{A_j} H_{ij} \frac{\cos\phi_i \cos\phi_j}{\pi r^2} dA_j dA_i$$

$$F_{A_j \to A_i} = \frac{1}{A_j} \int_{A_j} F_{dA_j \to A_i} dA_j = \frac{1}{A_j} \int_{A_j} \int_{A_i} H_{ji} \frac{\cos\phi_j \cos\phi_i}{\pi r^2} dA_i dA_j$$

$$F_{A_i \to A_j} A_i = F_{A_j \to A_i} A_j$$

Graphical interpretation



$$form\ factor = \frac{area\ of\ the\ projection}{area\ of\ the\ base\ circle}$$

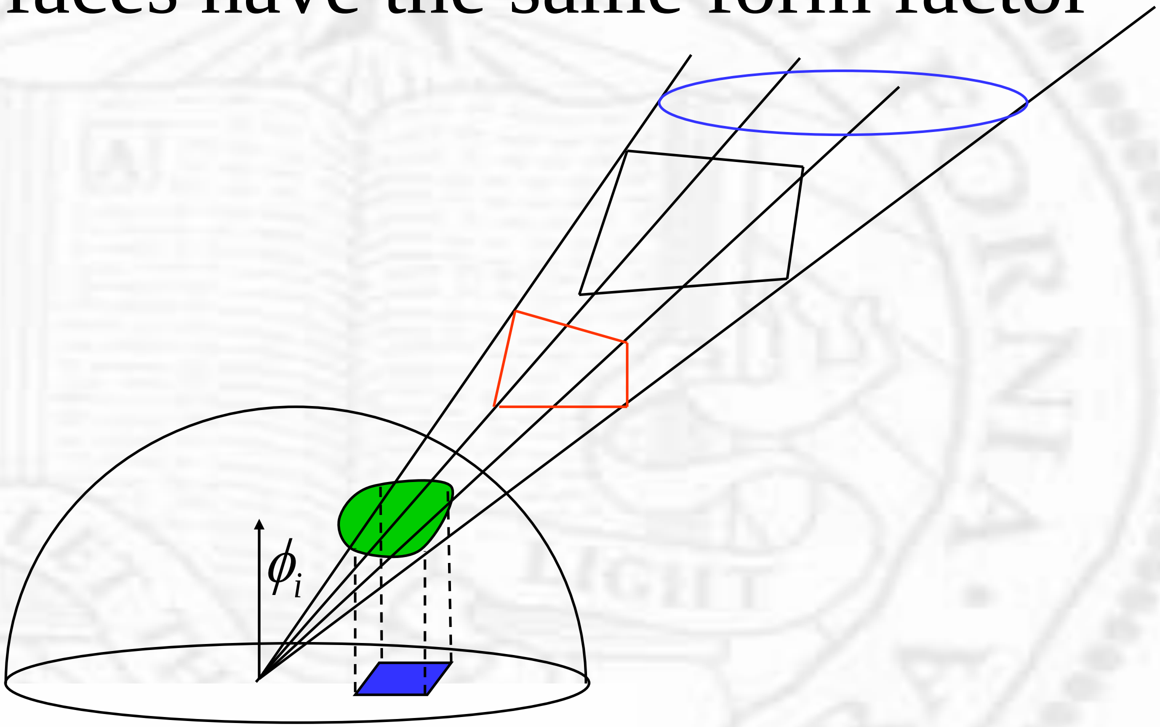
$$\text{projection onto the hemisphere } \frac{\cos\phi_j}{r^2}$$

$$\text{projection down on the the base } \cos\phi_i$$

$$\text{divided by the area of the base } \frac{1}{\pi}$$

Further simplification

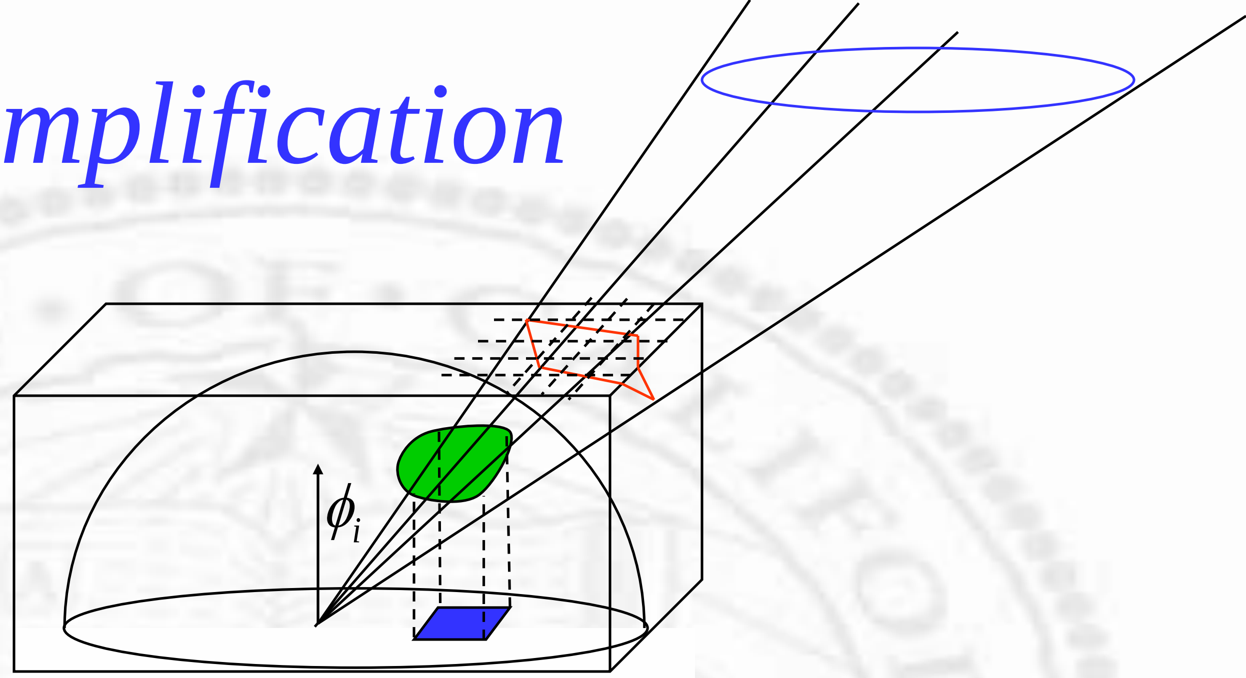
- As long as the same projection is produced, all these surfaces have the same form factor

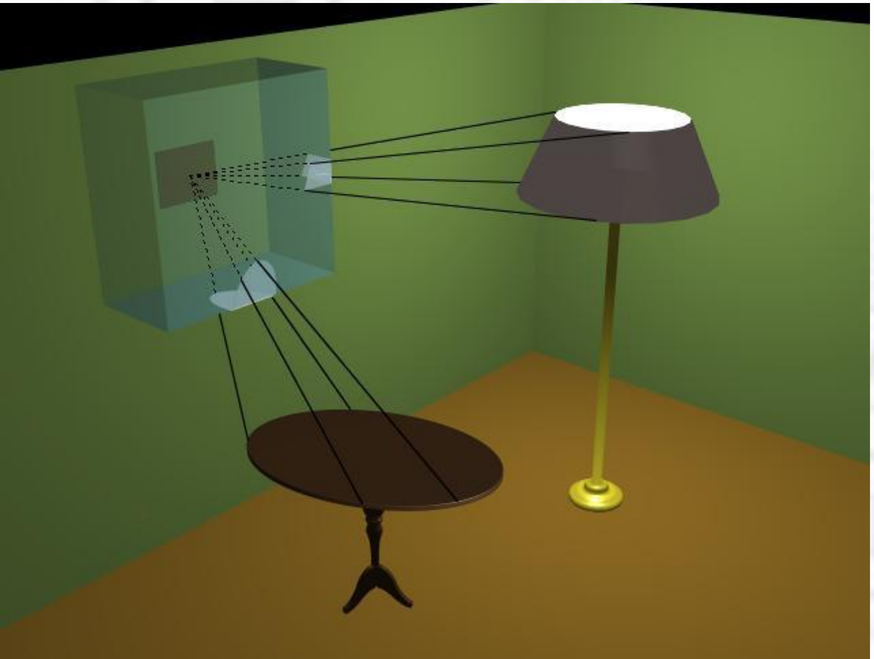


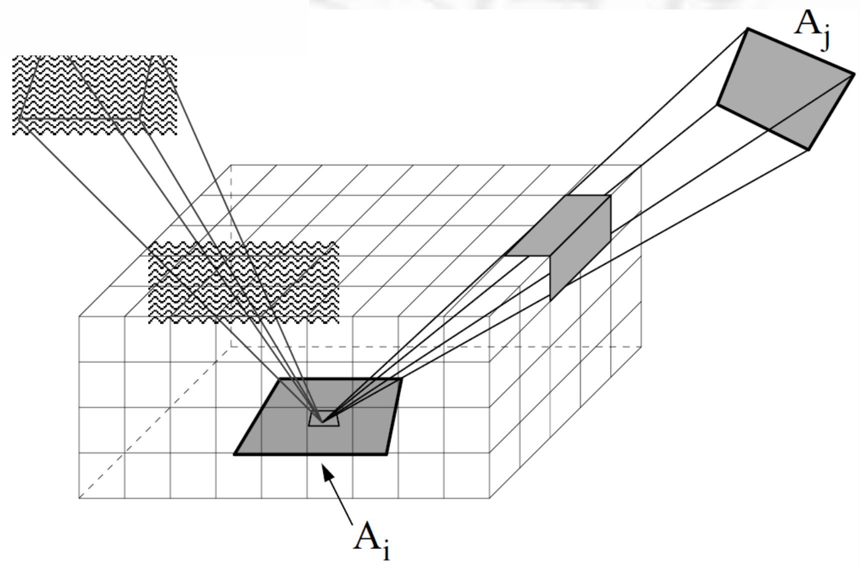
Simplification

- Instead of projecting onto a hemisphere, we can project onto a hemicube with planar surfaces (with traditional visible surface determination algorithm)
- The hemicube can be discretized and pixel radiosities tabulated in advance
- Then just count how many pixels a particular patch covers and add up individual radiosity values

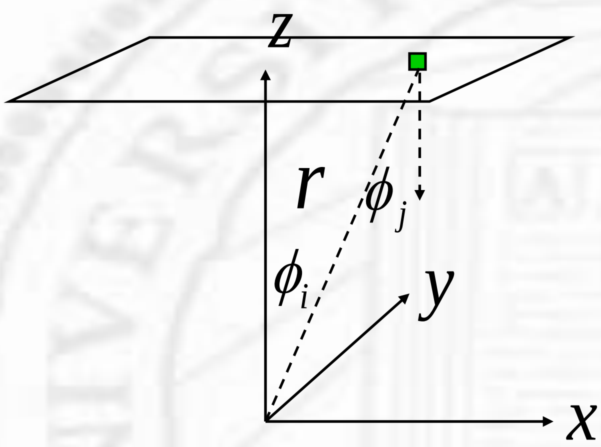
Simplification







Example



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

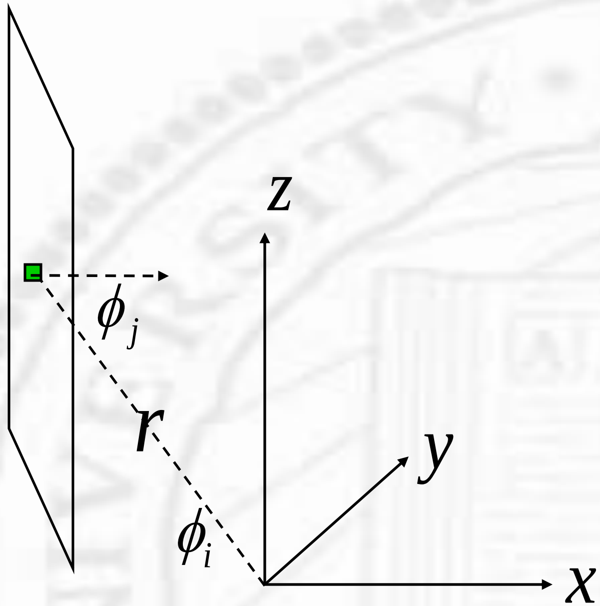
$$\cos\phi_i = \cos\phi_j = \frac{1}{\sqrt{x^2 + y^2 + 1}}$$

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

$$F_{dA_i \to dA_j} = H_{ij}\frac{\cos\phi_i \cos\phi_j}{\pi r^2}$$

$$H_{ij}\begin{cases} 1 & visible \\ 0 & not \end{cases}$$

More Example



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

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

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

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

Many Possible Generalizations

- Substructuring
  - Spatial refinement
- Progressive radiosity
  - Faster update
- Incremental radiosity
  - Temporal refinement

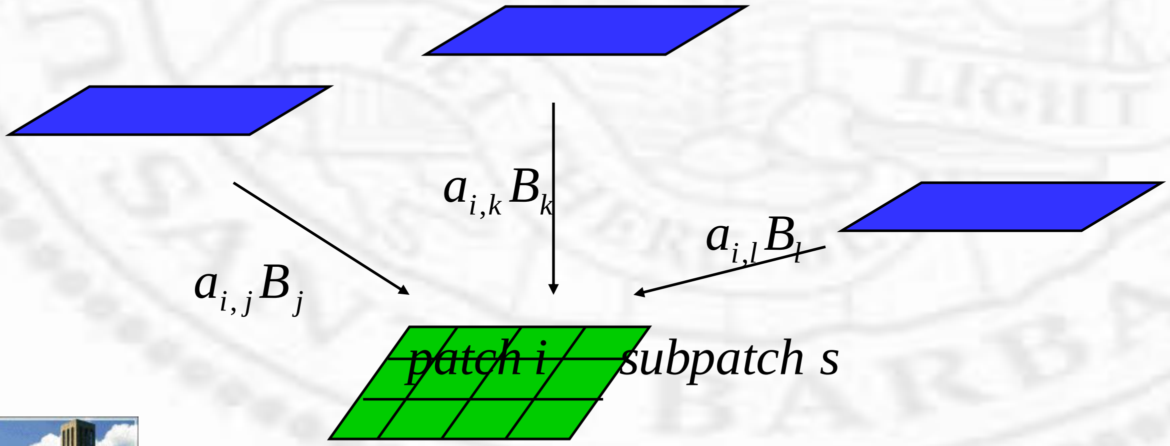
Substructuring

- At places with large radiosity changes
- Need smaller patches for better approximation
- Break one patch into *m* sub patches
  - introduce *m* more radiosity values
  - the radiosity matrix becomes $O((n+m)^2)$

- Instead
  - compute sub-patch form factors $F_{s \to j}$
  - update form factor of patch I $F_{i \to j} = \frac{1}{A_i} \sum_s F_{s \to j} A_s$
  - compute radiosity using original *nxn* equations
  - update radiosities of sub-patches

$$B_i = E_i + \rho_i \sum_j B_j F_{i \to j}$$

$$B_s = E_i + \rho_i \sum_j B_j F_{s \to j}$$



Progressive Radiosity

- For traditional radiosity solution, each iteration is of $O(n^2)$
- "Gathering" radiosity
  - n updates, one for each path
  - each update "gathers" the radiosity values of all n patches

$$a_{11}B_1^{(n)} = -(a_{12}B_2^{(n-1)} + a_{13}B_3^{(n-1)} + \cdots + a_{1n}B_n^{(n-1)}) + E_1$$

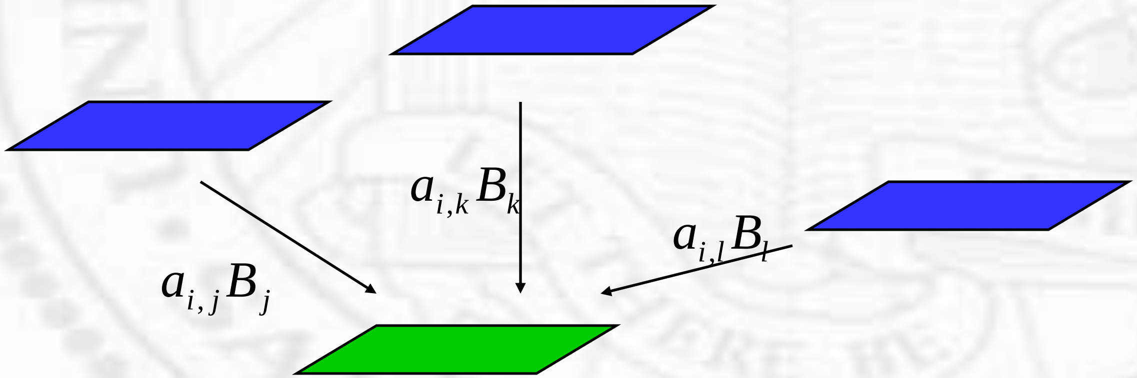
$$a_{22}B_2^{(n)} = -(a_{21}B_1^{(n)} + a_{23}B_3^{(n-1)} + \cdots + a_{2n}B_n^{(n-1)}) + E_2$$

$\ldots$

$$a_{ii}B_i^{(n)} = -(a_{i1}B_1^{(n)} + a_{i2}B_2^{(n)} + \cdots + a_{i,i-1}B_{i-1}^{(n)} + a_{i,i+1}B_{i+1}^{(n-1)} + \cdots + a_{in}B_n^{(n-1)}) + E_i$$

$\ldots$

$$a_{nn}B_n^{(n)} = -(a_{n1}B_1^{(n)} + a_{n2}B_2^{(n)} + \cdots + a_{n,n-1}B_{n-1}^{(n)}) + E_n$$



- 0 bounce 1 bounce 2 bounces

"Shooting" radiosity

- updates all n patches using the radiosity value of a single patch

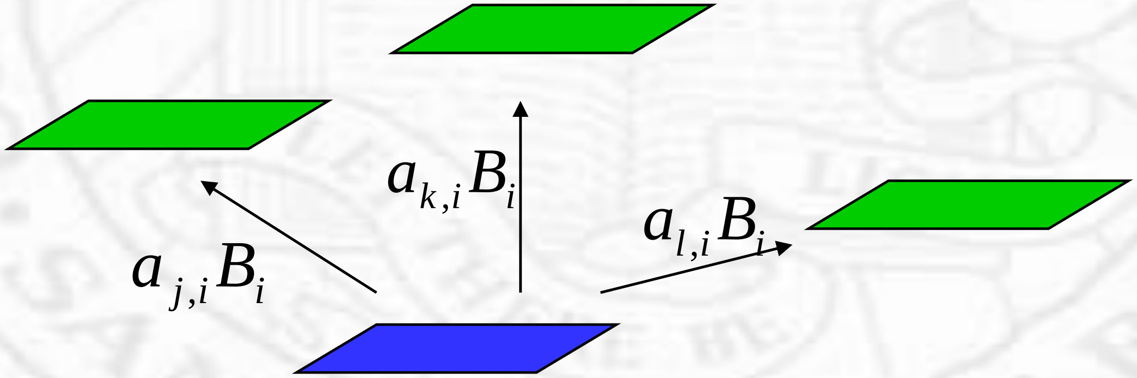
$$\begin{bmatrix} 1-\rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1i} & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1-\rho_2 F_{22} & \cdots & -\rho_2 F_{2i} & -\rho_2 F_{2n} \\ & & \cdots & & \\ & & \cdots & & \\ -\rho_n F_{n1} & -\rho_n F_{n2} & \cdots & -\rho_n F_{ni} & 1-\rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_i \\ E_n \end{bmatrix}$$

$$\Delta B_k = E_k \quad \forall k$$

$$a_{kk} B_k^{(n)} = B_k^{(n-1)} - \rho_k F_{k,i} \Delta B_i \quad \forall k$$

$$= B_k^{(n-1)} - \rho_k \frac{A_i}{A_k} F_{i,k} \Delta B_i$$

- One iteration involes *n O(1)* updates
- Form factors of one patch need be kept
- Only the part of radiosity that was not processed before need be "shot"



Trade-off

- Most accurate for diffuse lighting
- Photorealistic image
- Soft shadow, color bleeding

- Large computational effort
- Form factor computation

Combining Ray Tracing and Radiosity

- First compute view independent, global diffuse illumination with radiosity
- Then compute view dependent, global specular illumination using ray tracing

Example