Radiosity

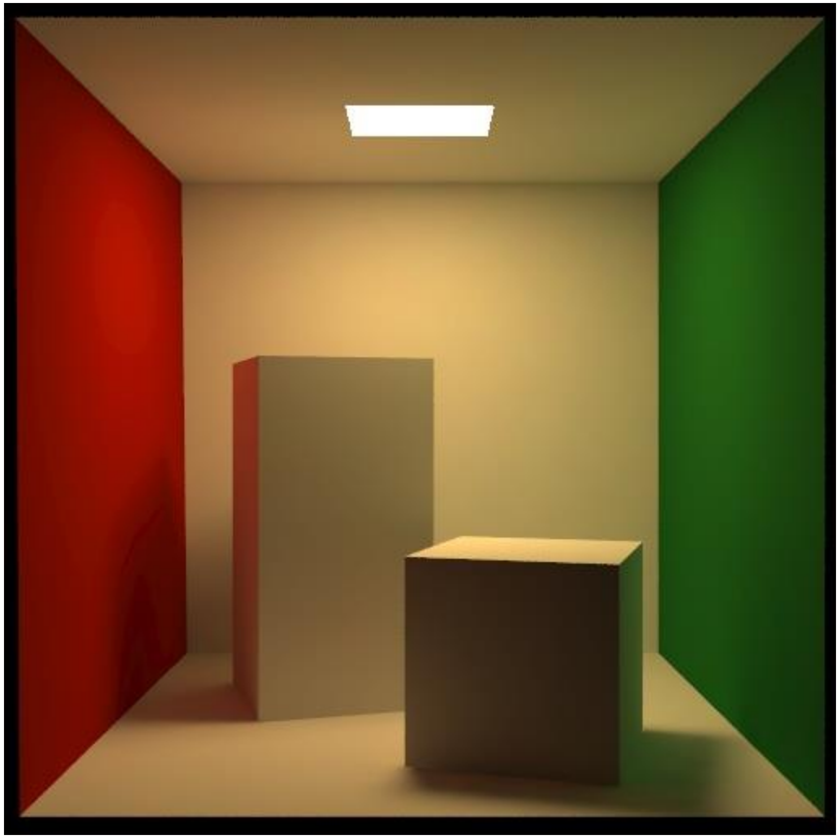
Dr. Scott Schaefer

TAMU
Computer
Science

Radiosity

Radiosity

- Physically based model for light interaction
- View independent lighting
- Accounts for indirect illumination
  - Color bleeding
  - Soft shadows



The Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l)\cos\theta_i d\omega_i$$

The Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l) \cos\theta_i \, d\omega_i$$

Outgoing radiance from surface at *p* in the direction *v*

The Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l) \cos\theta_i \, d\omega_i$$

Emitted radiance from surface at *p* in the direction *v*

The Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l) \cos\theta_i \, d\omega_i$$

BRDF of surface at *p*

The Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l) \cos\theta_i \, d\omega_i$$

Ray cast from *p* in the direction of *l*

The Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l) \cos\theta_i \, d\omega_i$$

Output radiance of intersected surface in the direction of -*l*

The Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l) \cos\theta_i \, d\omega_i$$

Angle between *l* and *n*

The Rendering Equation

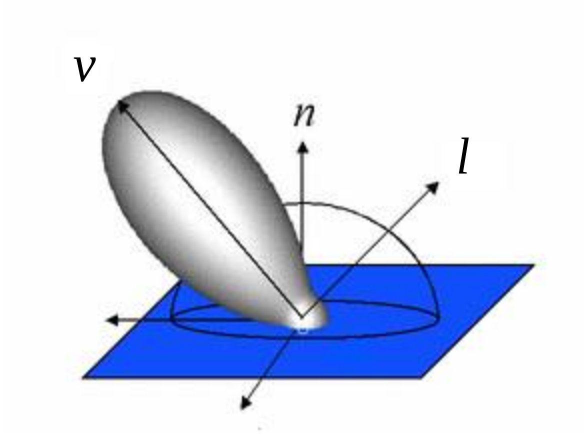
$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l)\cos\theta_i \, d\omega_i$$

Integral about hemisphere centered at *p*

BRDF's

Bidirectional **R**eflectance **D**istribution **F**unction

- Determines amount of incoming light from direction *l* reflected from surface in the direction of *v*



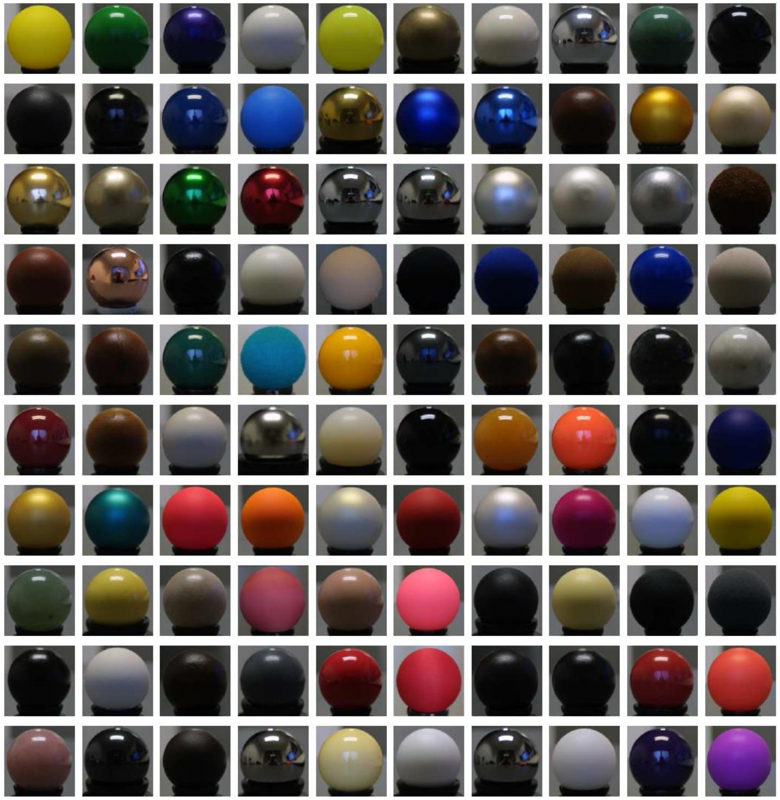
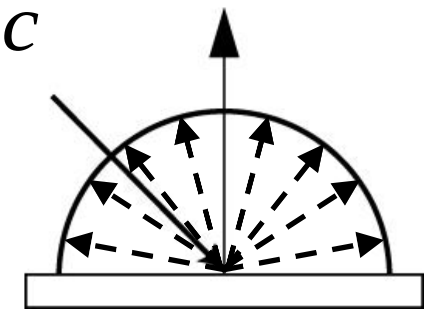


Image taken from "A Data-Driven Reflectance Model"

Discretizing the Rendering Equation

- Assume perfectly diffuse surfaces

  $f(l, v) = c$



- Discretize space into patches
- Color is constant per patch

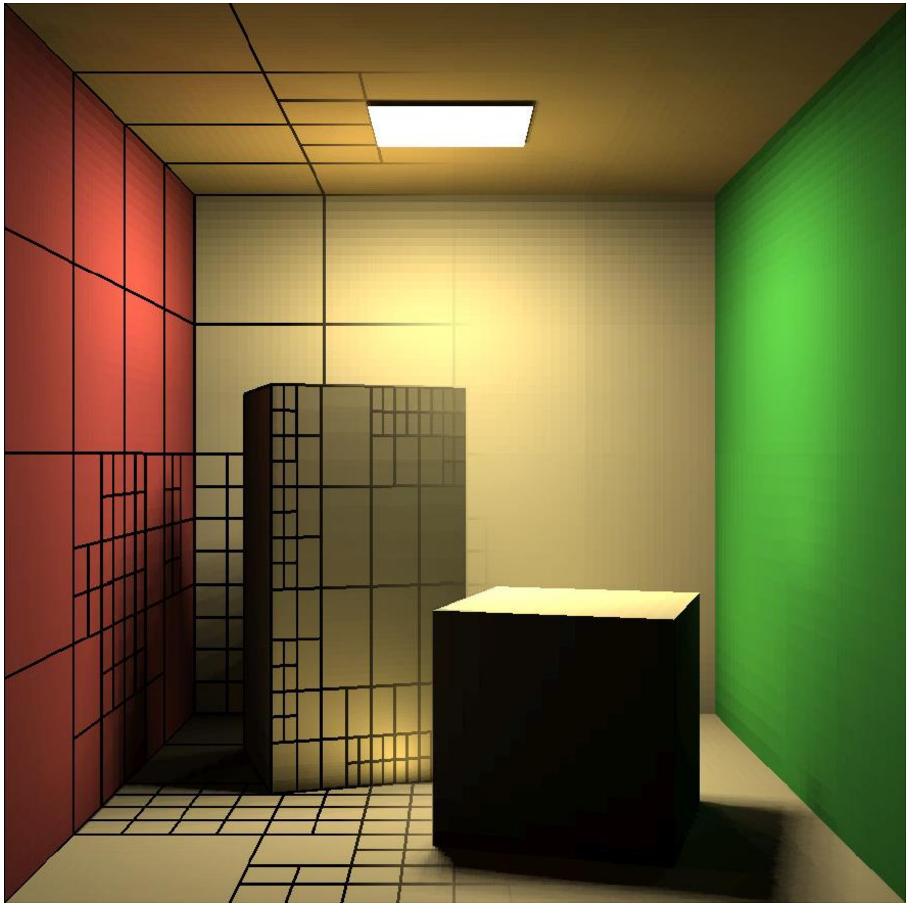


Image taken from "Radiosity on Graphics Hardware"

Discretizing the Rendering Equation

$$L_o(p,v) = L_e(p,v) + \int_\Omega f(l,v) L_o(r(p,l),-l) \cos\theta_i \, d\omega_i$$

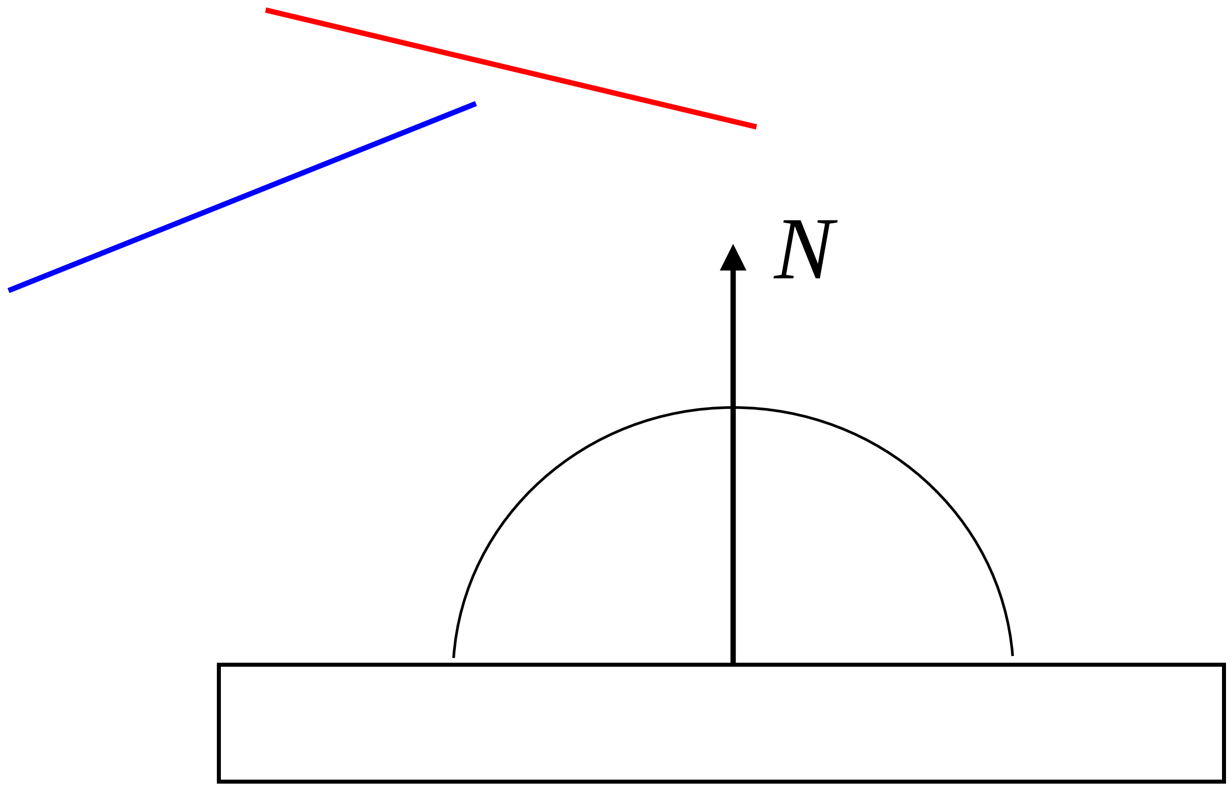
Discretizing the Rendering Equation

$$L_i = L_{i,e} + \int_{\Omega} c_i L_j h_{i,j} \cos\theta_i \, d\omega_i$$

$$h_{i,j} = \begin{cases} 1 & \text{patch } i \text{ is visible to patch } j \text{ along } l \\ 0 & \text{patch } i \text{ is not visible to patch } j \text{ along } l \end{cases}$$

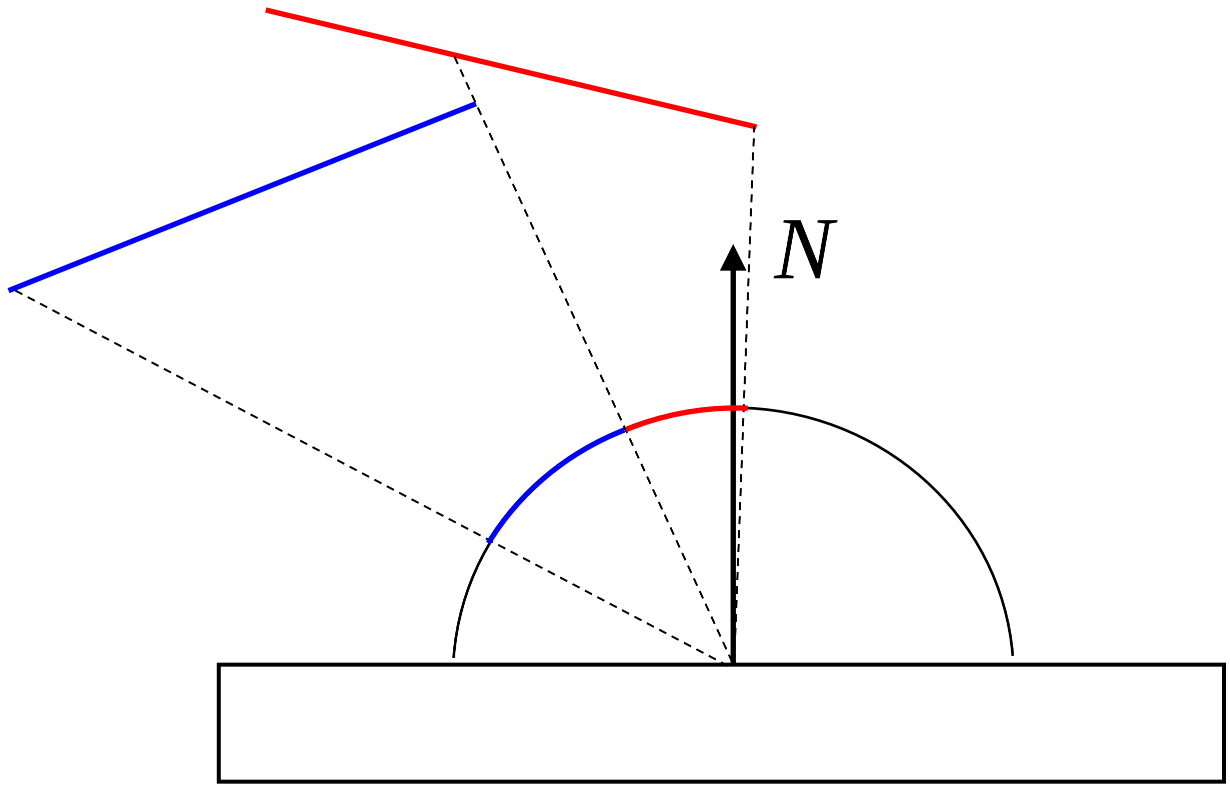
Geometric Computation of Form Factors

$$L_i = L_{i,e} + \int_{\Omega} c_i L_j h_{i,j} \cos\theta_i d\omega_i$$



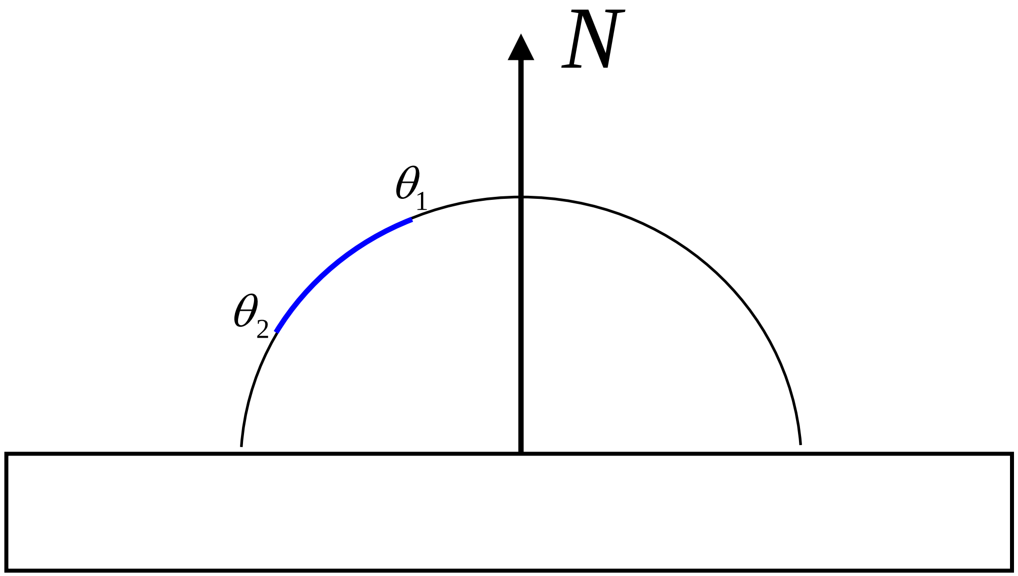
Geometric Computation of Form Factors

$$L_i = L_{i,e} + \int_\Omega c_i L_j h_{i,j} \cos\theta_i d\omega_i$$



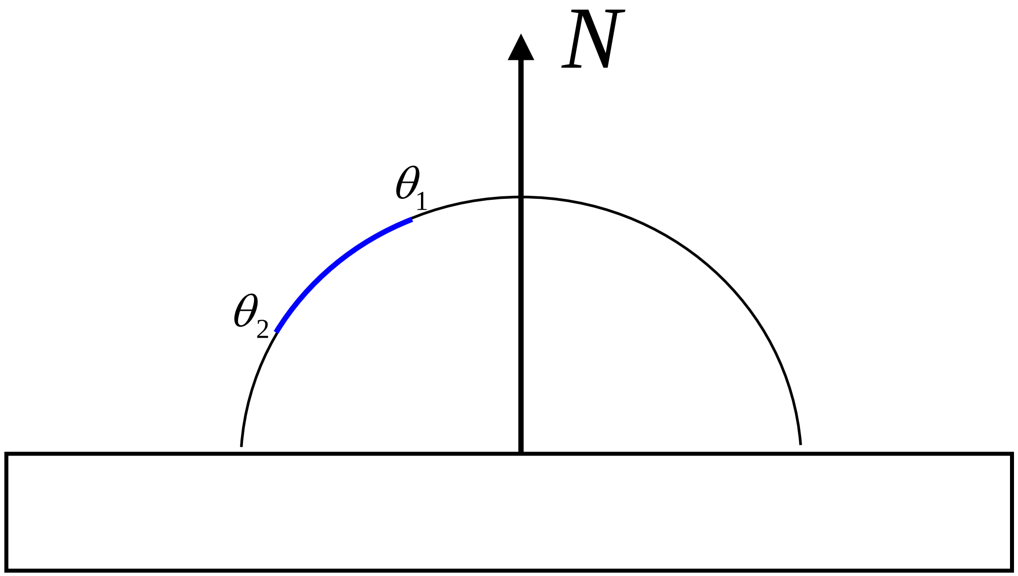
Geometric Computation of Form Factors

$$L_i = L_{i,e} + \int_{\theta_1}^{\theta_2} c_i L_j \cos\theta d\theta$$



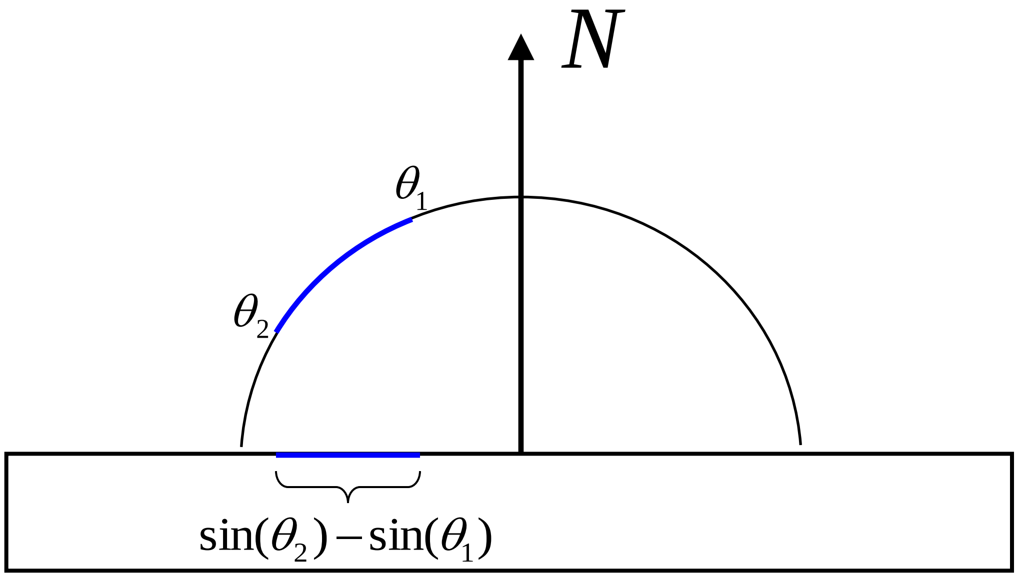
Geometric Computation of Form Factors

$$L_i = L_{i,e} + c_i L_j \left(\sin\theta_2 - \sin\theta_1\right)$$

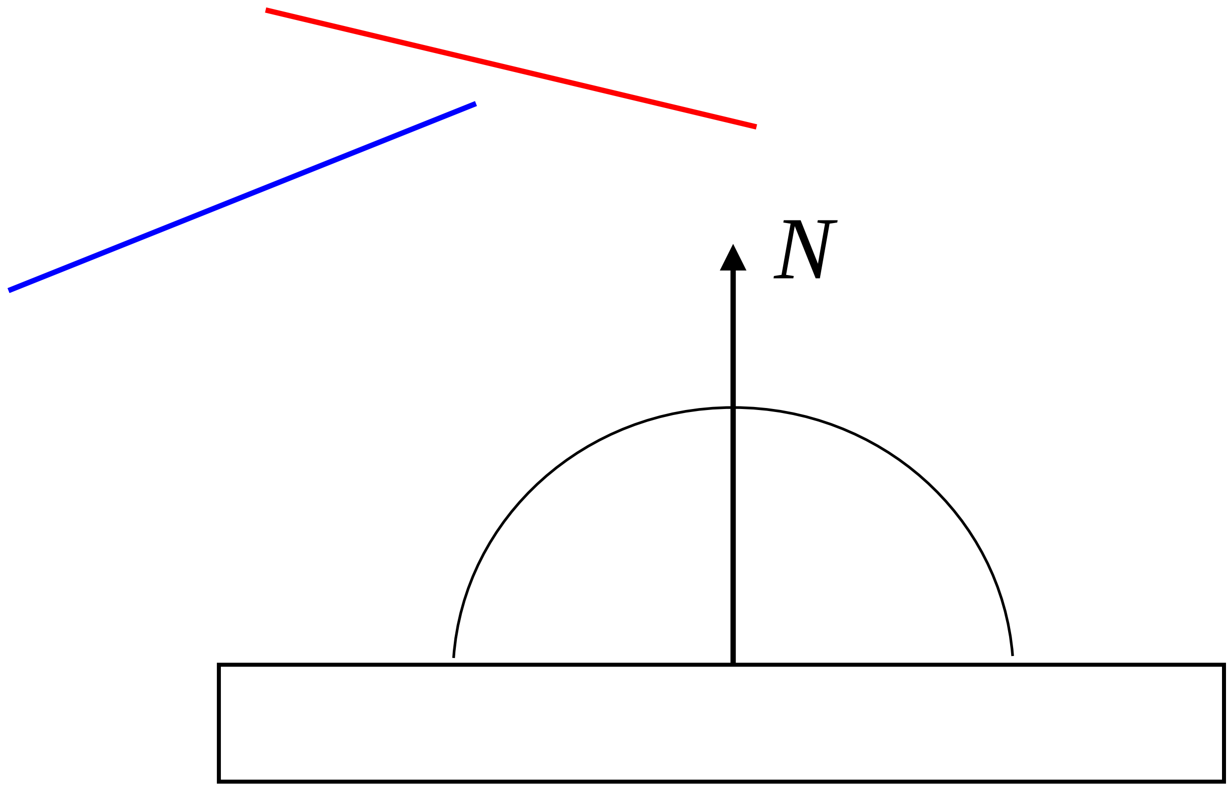


Geometric Computation of Form Factors

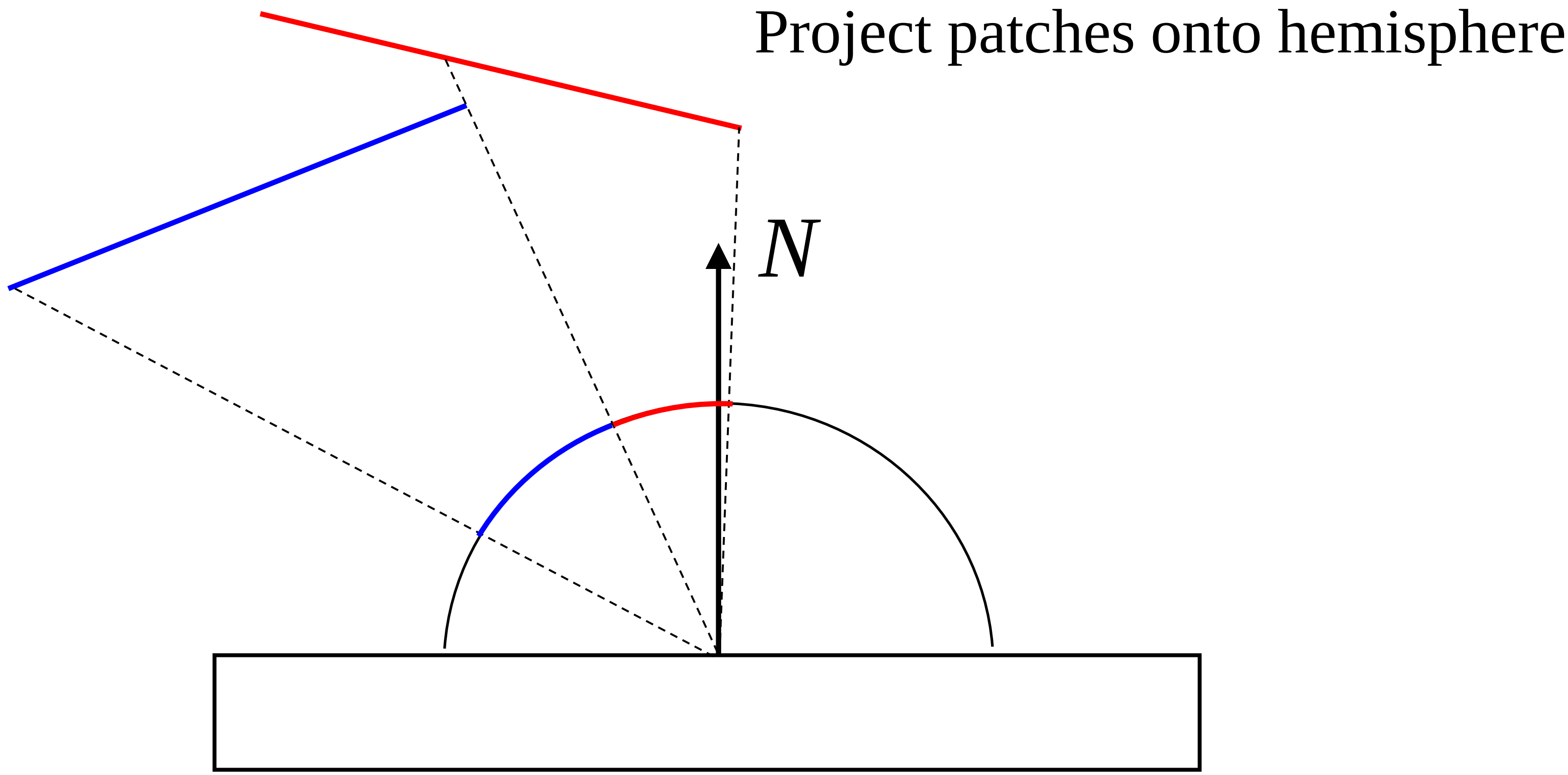
$$L_i = L_{i,e} + c_i L_j \left(\sin\theta_2 - \sin\theta_1 \right)$$



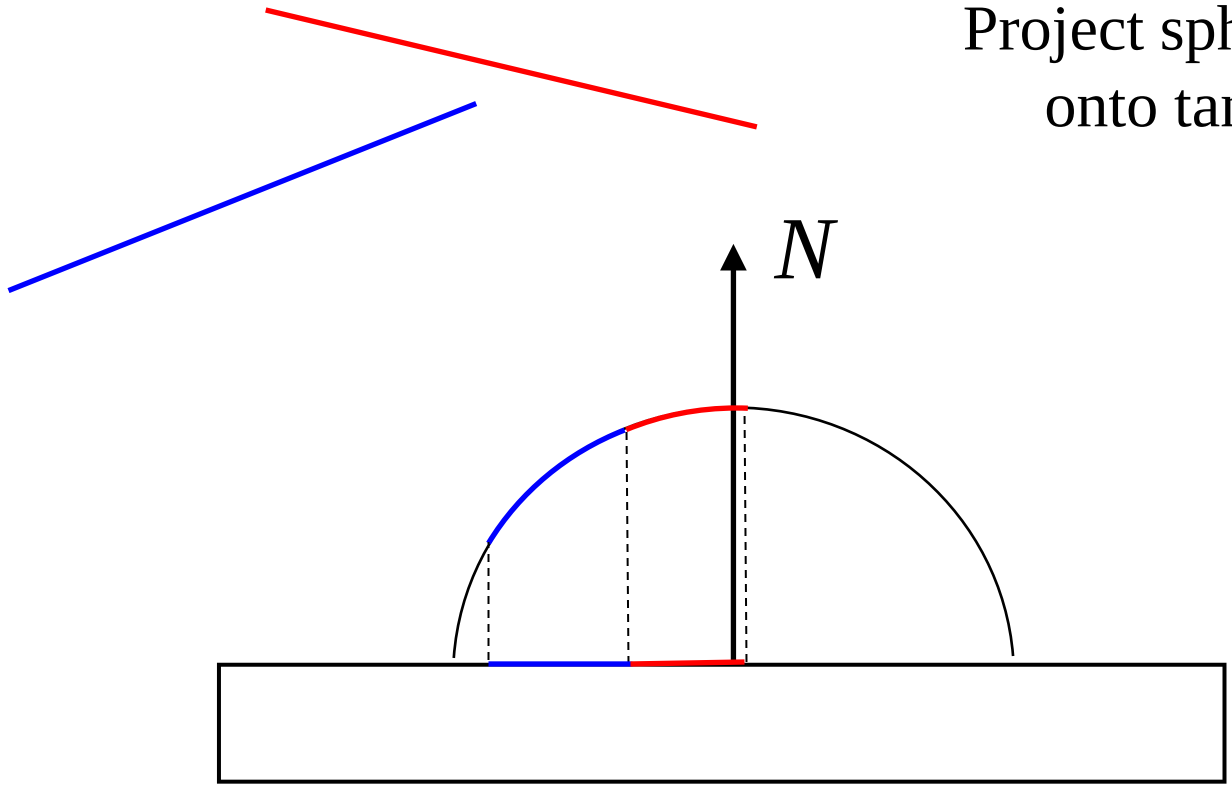
Geometric Computation of Form Factors



Geometric Computation of Form Factors

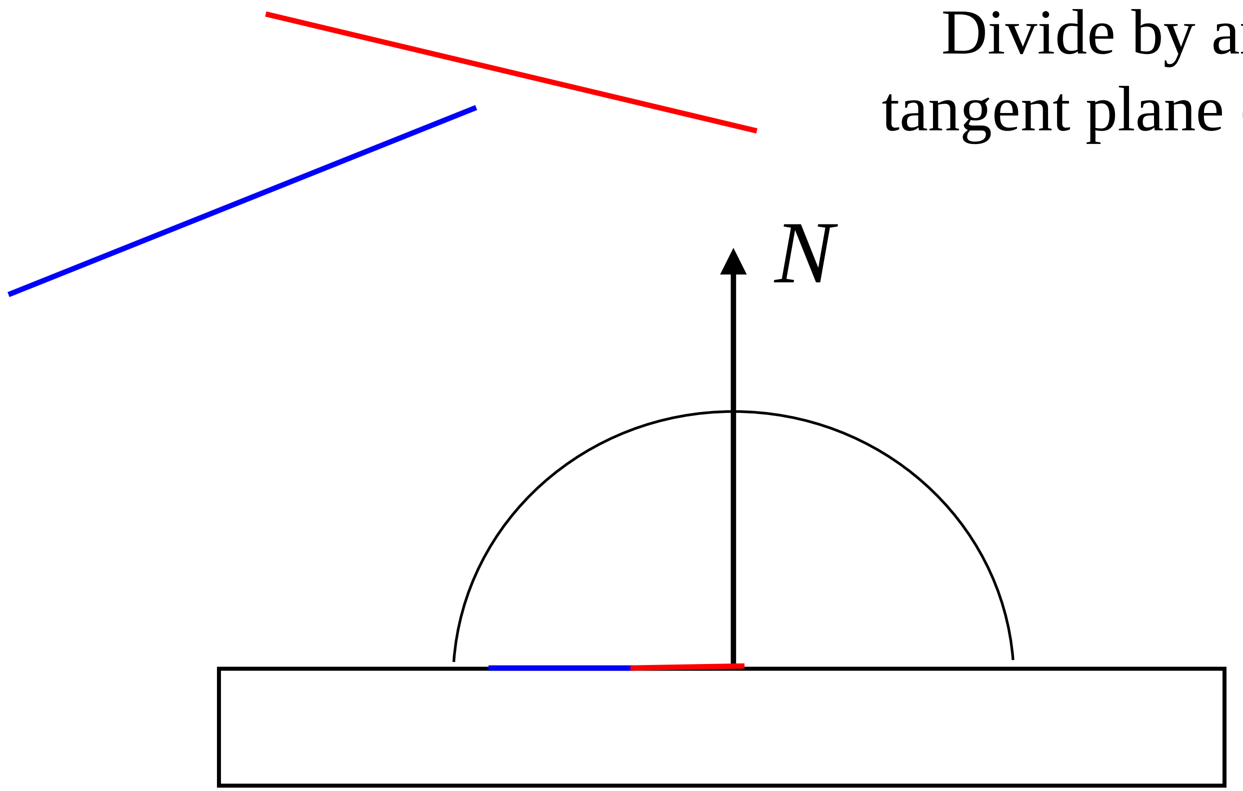


Geometric Computation of Form Factors



Project spherical patches
onto tangent plane

Geometric Computation of Form Factors

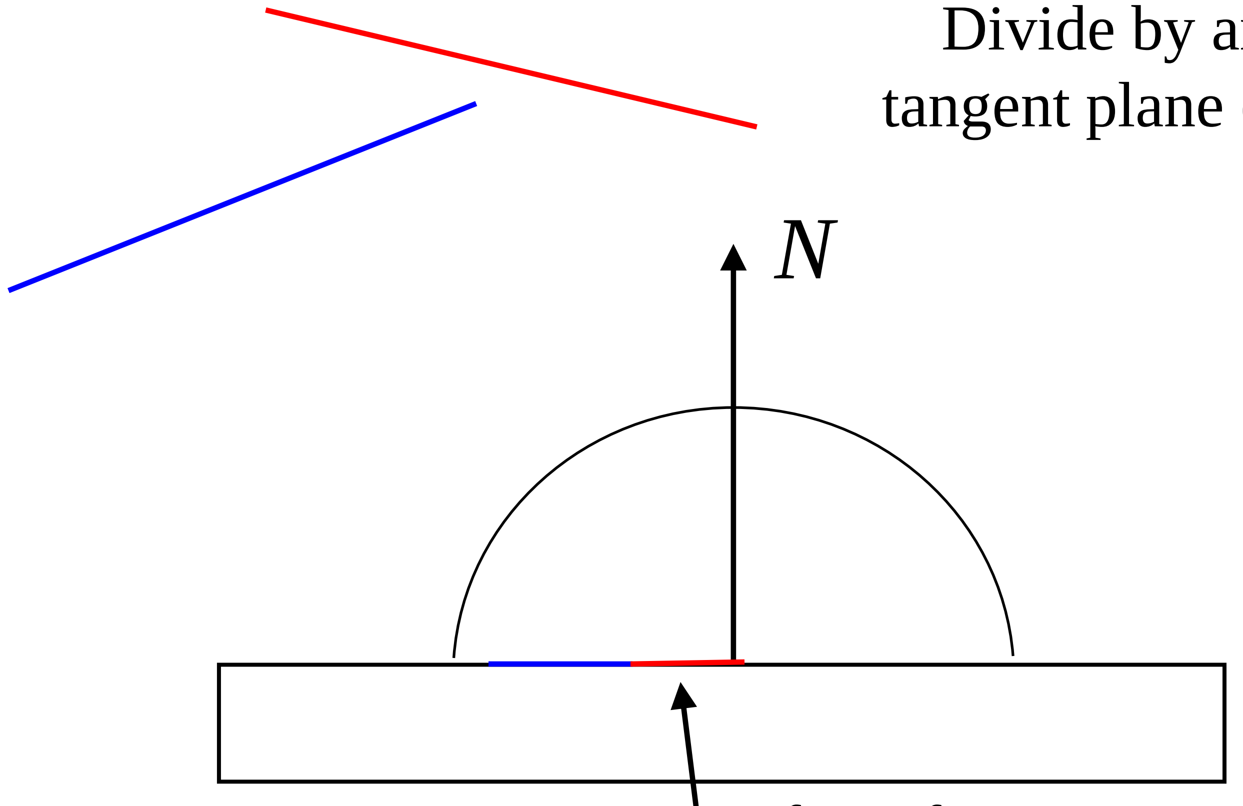


Divide by area of disc in
tangent plane (π for surfaces)

Geometric Computation of Form Factors

$$L_i = L_{i,e} + \sum_j c_i L_j F_{i,j}$$

Divide by area of disc in tangent plane (π for surfaces)



$F_{i,j}$ = form factor

Matrix Computation of Radiosity

$$L_i = L_{i,e} + \sum_j c_i L_j F_{i,j}$$

$$\begin{pmatrix} 1-c_1F_{1,1} & -c_1F_{1,2} & \cdots & -c_1F_{1,n} \\ -c_2F_{2,1} & 1-c_2F_{2,2} & \cdots & -c_2F_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ -c_nF_{n,1} & -c_nF_{n,2} & \cdots & 1-c_nF_{n,n} \end{pmatrix} \begin{pmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{pmatrix} = \begin{pmatrix} L_{e,1} \\ L_{e,2} \\ \vdots \\ L_{e,n} \end{pmatrix}$$

Matrix Computation of Radiosity

$$L_i = L_{i,e} + \sum_j c_i L_j F_{i,j}$$

$$\begin{pmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{pmatrix} = \begin{pmatrix} L_{e,1} \\ L_{e,2} \\ \vdots \\ L_{e,n} \end{pmatrix} + \begin{pmatrix} c_1 F_{1,1} & c_1 F_{1,2} & \cdots & c_1 F_{1,n} \\ c_2 F_{2,1} & c_2 F_{2,2} & \cdots & c_2 F_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ c_n F_{n,1} & c_n F_{n,2} & \cdots & c_n F_{n,n} \end{pmatrix} \begin{pmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{pmatrix}$$

Matrix Computation of Radiosity

$$L_i = L_{i,e} + \sum_j c_i L_j F_{i,j}$$

$$\begin{pmatrix} L^i{}_1 \\ L^i{}_2 \\ \vdots \\ L^i{}_n \end{pmatrix} = \begin{pmatrix} L_{e,1} \\ L_{e,2} \\ \vdots \\ L_{e,n} \end{pmatrix} + \begin{pmatrix} c_1 F_{1,1} & c_1 F_{1,2} & \cdots & c_1 F_{1,n} \\ c_2 F_{2,1} & c_2 F_{2,2} & \cdots & c_2 F_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ c_n F_{n,1} & c_n F_{n,2} & \cdots & c_n F_{n,n} \end{pmatrix} \begin{pmatrix} L^{i-1}{}_1 \\ L^{i-1}{}_2 \\ \vdots \\ L^{i-1}{}_n \end{pmatrix}$$

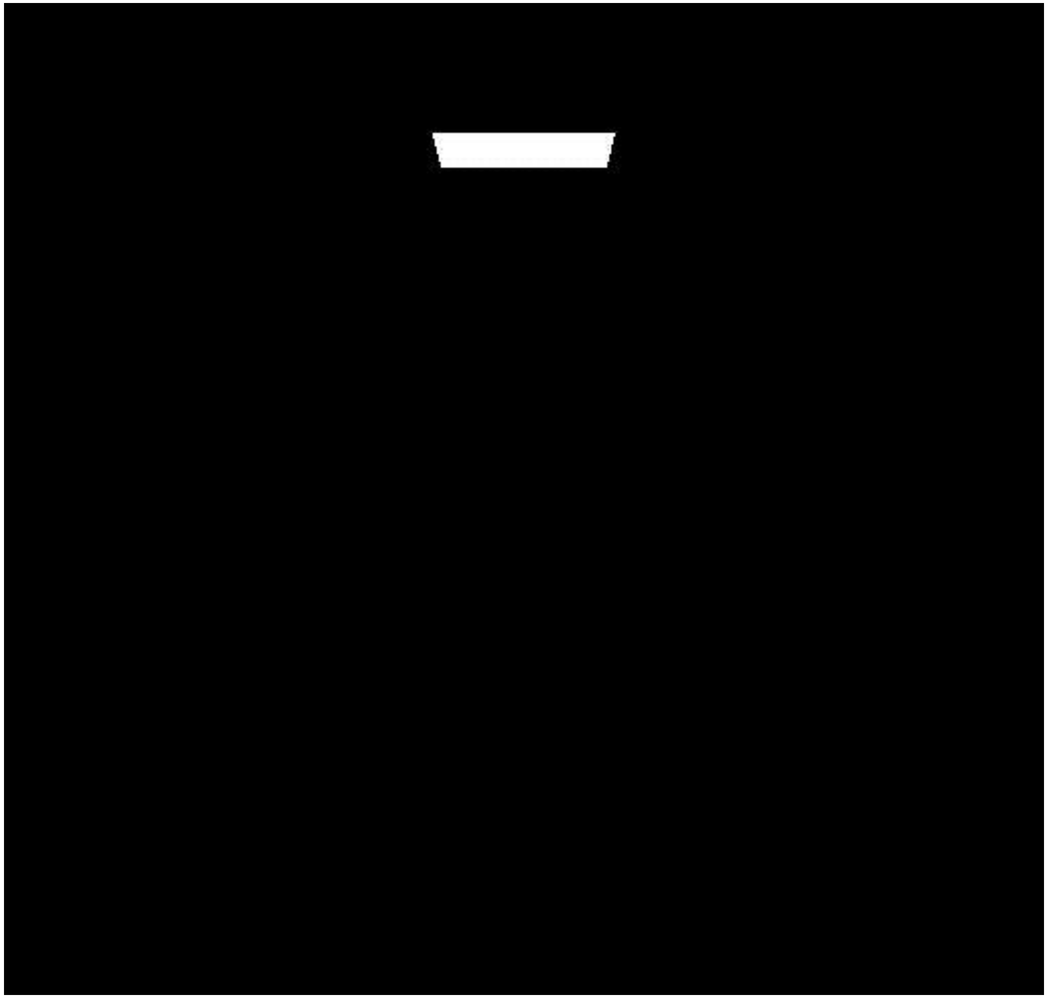
Jacobi iteration

Matrix Computation of Radiosity

$$L_i = L_{i,e} + \sum_j c_i L_j F_{i,j}$$

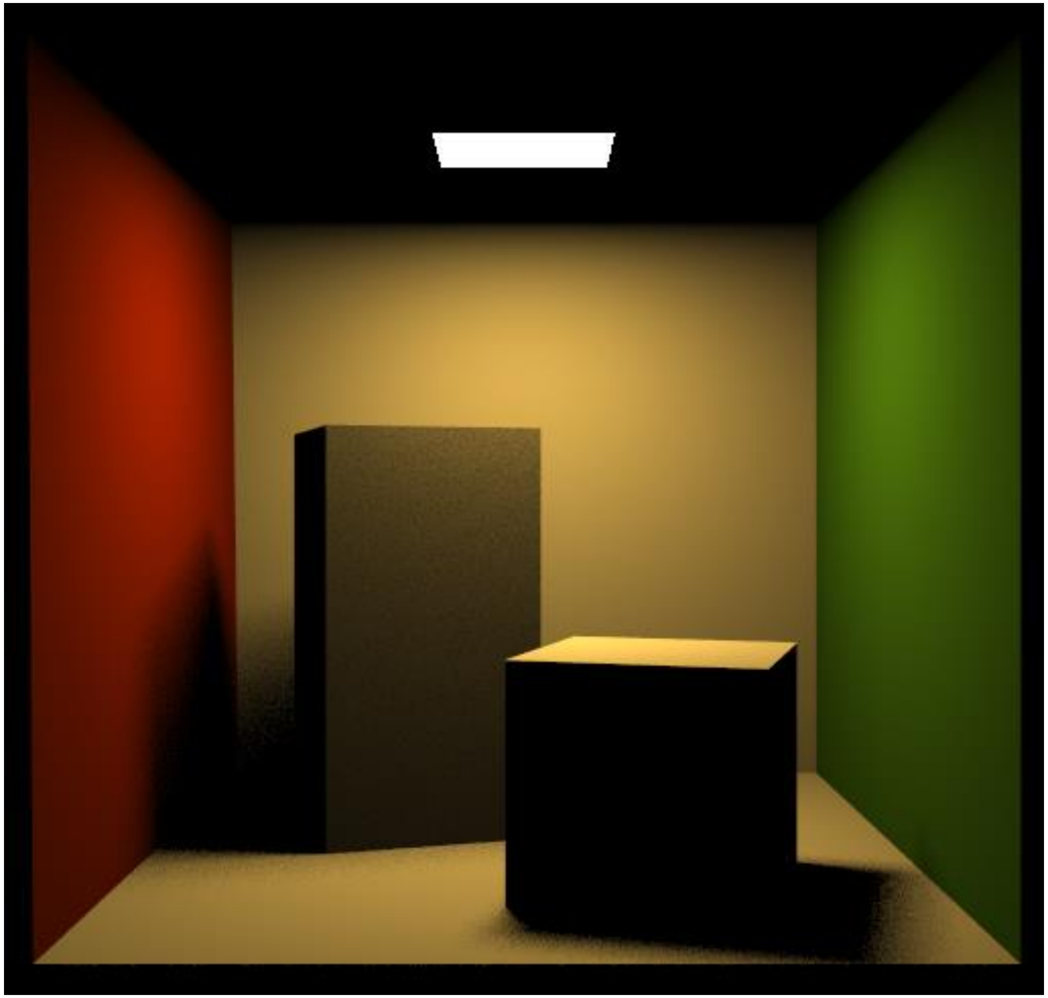
$$L^i = L_e + K\, L^{i-1}$$

Matrix Computation of Radiosity



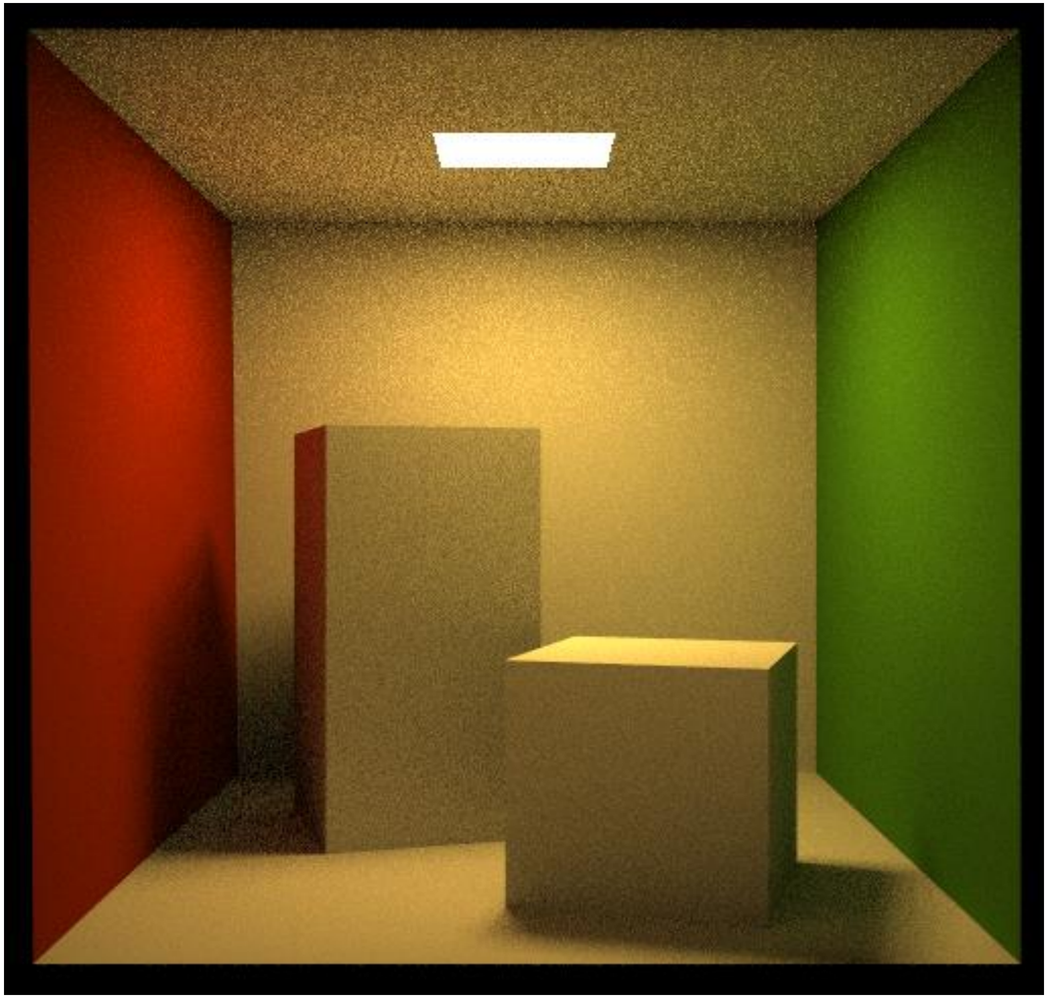
L_e

Matrix Computation of Radiosity



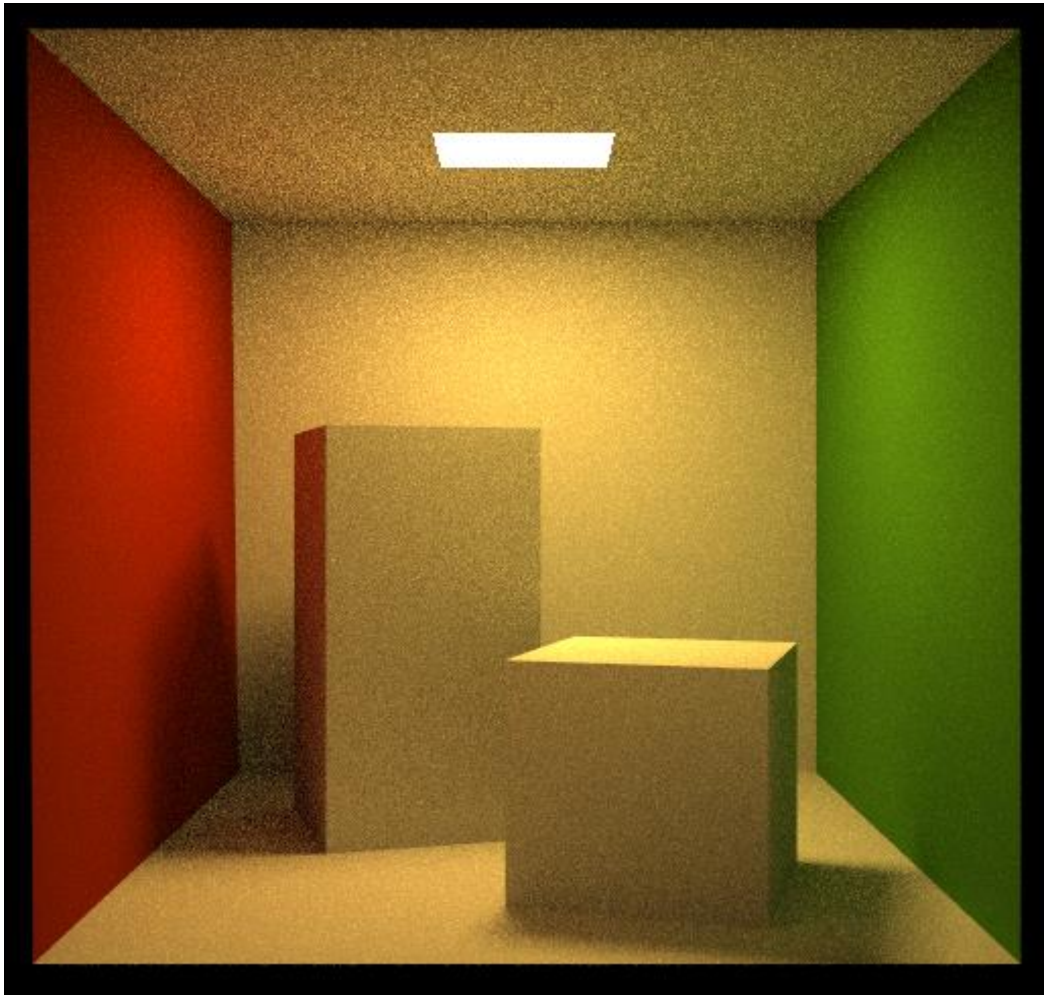
$$L_e + K\,L_e$$

Matrix Computation of Radiosity



$$L_e + K\,L_e + K^2 L_e$$

Matrix Computation of Radiosity



$$L_e + K\,L_e + K^2 L_e + K^3 L_e$$

Radiosity Examples

Image taken from http://www.9jcg.com/tutorials/jason_jacobs/radiosity_01.jpg

Radiosity Examples

Radiosity Examples

Image taken from www.cs.dartmouth.edu/~spl/Academic/ComputerGraphics/Fall2004/radiosityExample.jpg

Advantages of Radiosity

- Global illumination method: modeling diffuse inter-reflection
- Color bleeding: a red wall next to a white one casts a reddish glow on the white wall
- Soft shadows: an area light source casts a soft shadow from a polygon
- No ambient hack
- View independent: assigns a brightness to every surface

Disadvantages of Radiosity

- Radiation is uniform in all directions
- Radiosity is piecewise constant
- No surface is transparent or translucent
- Must determine how to subdivide shapes into small enough patches