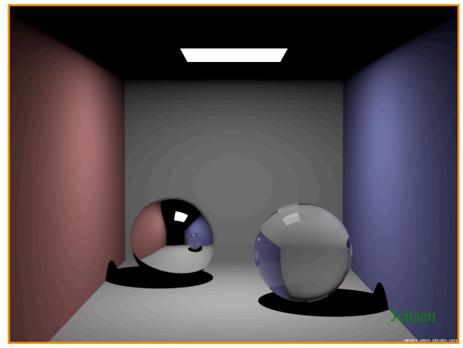
Global Illumination

Adam Finkelstein
Princeton University
COS 526, Fall 2005

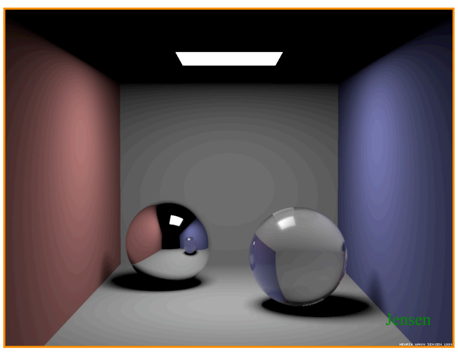
Path Types



Ray tracing

Henrik Wann Jensen

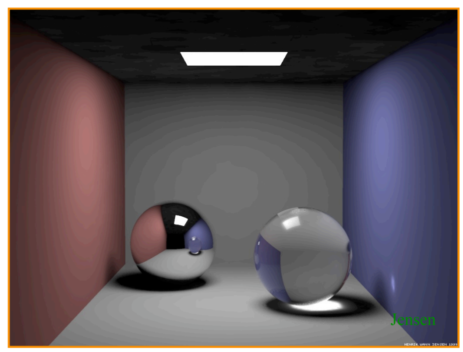
Path Types



+ soft shadows

Henrik Wann Jensen

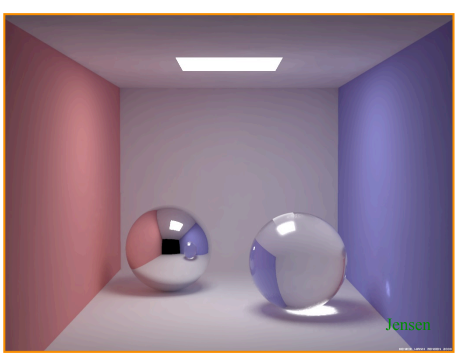
Path Types



+ caustics

Henrik Wann Jensen

Path Types

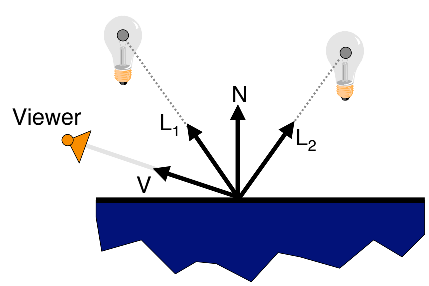


+ indirect diffuse illumination

Henrik Wann Jensen

Direct Illumination

- Multiple light sources:



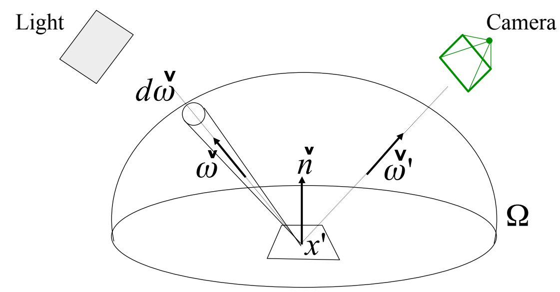
$$I = I_E + K_A I_{AL} + \sum_i (K_D (N \bullet L_i) I_i + K_S (V \bullet R_i)^n I_i)$$

Overview

- Global illumination
  - Rendering equation
- Solution methods
  - OpenGL
  - Ray tracing
  - Path tracing
  - Radiosity
- Path types
  - L(S|D)*E

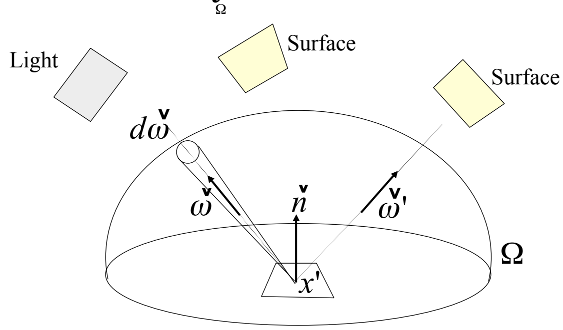
Direct Illumination

$$L_o(x',\check{\omega}') = L_e(x',\check{\omega}') + \int_{\Omega_L} f_r(x',\check{\omega},\check{\omega}') L_i(x',\check{\omega})(\check{\omega} \bullet \check{n}) d\check{\omega}$$



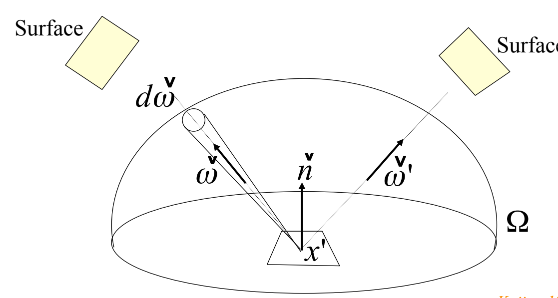
Global Illumination

$$L_o(x',\check{\omega}') = L_e(x',\check{\omega}') + \int_{\Omega} f_r(x',\check{\omega},\check{\omega}') L_i(x',\check{\omega})(\check{\omega} \bullet \check{n}) d\check{\omega}$$



Rendering Equation

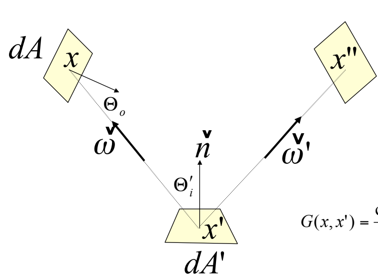
$$L_o(x',\check{\omega}') = L_e(x',\check{\omega}') + \int_{\Omega} f_r(x',\check{\omega},\check{\omega}') L_i(x',\check{\omega})(\check{\omega} \bullet \check{n}) d\check{\omega}$$



Kajiya 1986

Rendering Equation (2)

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



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

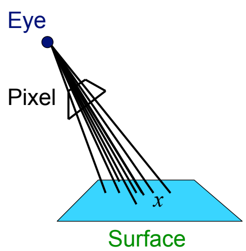
Kajiya 1986

Photorealistic Rendering

- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

Photorealistic Rendering

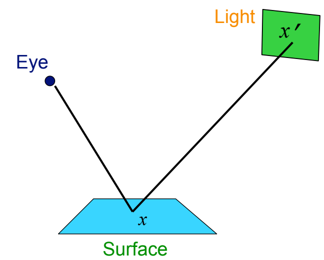
- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics



$$L_P = \int_S L(x \to e)dA$$

Photorealistic Rendering

- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics



$$L(x,\vec{w}) = L_e(x,x \to e) + \int_S f_r(x,x' \to x, x \to e) L(x' \to x) V(x,x') G(x,x') dA$$

Photorealistic Rendering

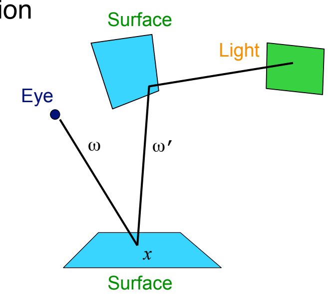
- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

Herf

$$L(x,\vec{w}) = L_e(x,x \to e) + \int_S f_r(x,x' \to x, x \to e) L(x' \to x) V(x,x') G(x,x') dA$$

Photorealistic Rendering

- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics



$$L_o(x,\vec{w}) = L_e(x,\vec{w}) + \int_\Omega f_r(x,\vec{w}',\vec{w}) L_i(x,\vec{w}')(\vec{w}' \bullet \vec{n}) d\vec{w}$$

Photorealistic Rendering

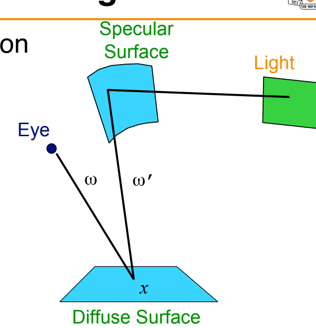
- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

Debevec

$$L_o(x,\vec{w}) = L_e(x,\vec{w}) + \int_\Omega f_r(x,\vec{w}',\vec{w}) L_i(x,\vec{w}')(\vec{w}' \bullet \vec{n}) d\vec{w}$$

Photorealistic Rendering

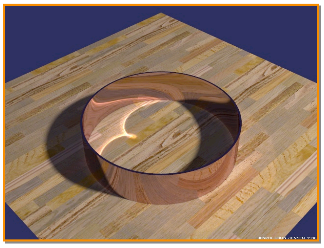
- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics



$$L_o(x,\vec{w}) = L_e(x,\vec{w}) + \int_\Omega f_r(x,\vec{w}',\vec{w}) L_i(x,\vec{w}')(\vec{w}' \bullet \vec{n}) d\vec{w}$$

Photorealistic Rendering

- Rendering = integration
  - Antialiasing
  - Soft shadows
  - Indirect illumination
  - Caustics

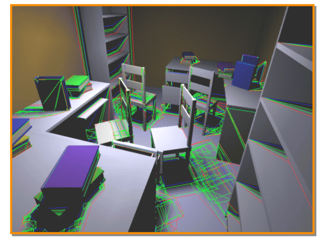


Jensen

$$L_o(x,\vec{w}) = L_e(x,\vec{w}) + \int_{\Omega} f_r(x,\vec{w}',\vec{w}) L_i(x,\vec{w}')(\vec{w}' \bullet \vec{n}) d\vec{w}$$

Challenge

- Rendering integrals are difficult to evaluate
  - Recursion: L = f(L)
  - Multiple dimensions
  - Discontinuities
    - » Partial occluders
    - » Highlights
    - » Caustics



Drettakis

$$L(x,\vec{w}) = L_e(x,x \to e) + \int_S f_r(x,x' \to x, x \to e) L(x' \to x) V(x,x') G(x,x') dA$$

Challenge

- Rendering integrals are difficult to evaluate
  - Recursion: L = f(L)
  - Multiple dimensions
  - Discontinuities
    - » Partial occluders
    - » Highlights
    - » Caustics

Jensen

$$L(x,\vec{w}) = L_e(x,x \to e) + \int_S f_r(x,x' \to x, x \to e) L(x' \to x) V(x,x') G(x,x') dA$$

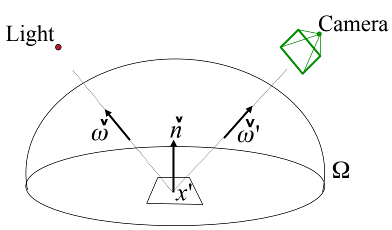
Overview

- Global illumination
  - Rendering equation
- Solution methods
  - OpenGL
  - Ray tracing
  - Path tracing
  - Radiosity
- Path types
  - L(S|D)*E

OpenGL

$$L_o(x',\vec{\omega}') = L_e(x',\vec{\omega}') + \int_{\Omega} f_r(x',\vec{\omega},\vec{\omega}') L_i(x',\vec{\omega})(\vec{\omega} \bullet \vec{n}) d\vec{\omega}$$

Assume direct illumination from point lights and ignore visibility

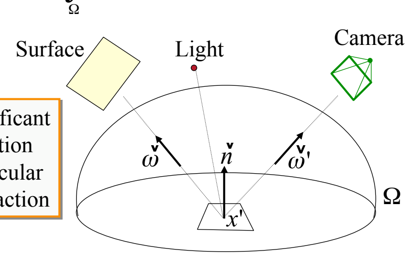


$$L_o(x',\vec{\omega}') = L_e(x',\vec{\omega}') + \sum_{i=1}^{nlights} f_r(x',\vec{\omega},\vec{\omega}') L_i(x',\vec{\omega})(\vec{\omega} \bullet \vec{n})$$

Ray Tracing

$$L_o(x',\vec{\omega}') = L_e(x',\vec{\omega}') + \int_{\Omega} f_r(x',\vec{\omega},\vec{\omega}') L_i(x',\vec{\omega})(\vec{\omega} \bullet \vec{n}) d\vec{\omega}$$

Assume only significant indirect illumination due to perfect specular reflection and refraction

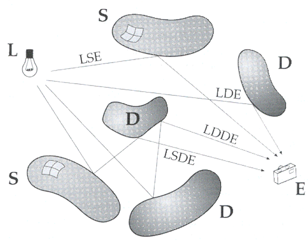


$$L_o(x',\vec{\omega}') = L_e(x',\vec{\omega}') + \sum_{i=1}^{nlights} f_r(x',\vec{\omega},\vec{\omega}') L_i(x',\vec{\omega})(\vec{\omega} \bullet \vec{n}) + specular$$

Monte Carlo Path Tracing

$$L_o(x',\vec{\omega}') = L_e(x',\vec{\omega}') + \int_{\Omega} f_r(x',\vec{\omega},\vec{\omega}')L_i(x',\vec{\omega})(\vec{\omega}\bullet\vec{n})d\vec{\omega}$$

Estimate integral for each pixel by random sampling

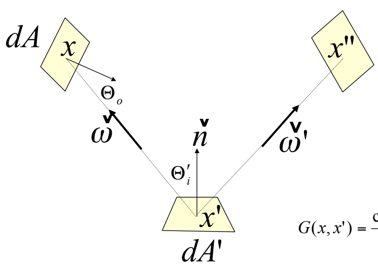


Also:
- Depth of field
- Motion blur
- etc.

Indirect Diffuse Illumination

Rendering Equation

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



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

Kajiya 1986

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

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

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

Convert to Radiosities

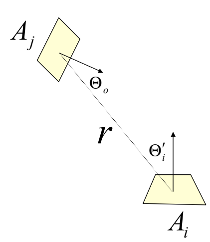
$$B = \int_\Omega L_o \cos\theta d\omega \qquad L = \frac{B}{\pi}$$

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

Radiosity Approximation

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

Discretize the 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

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

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

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

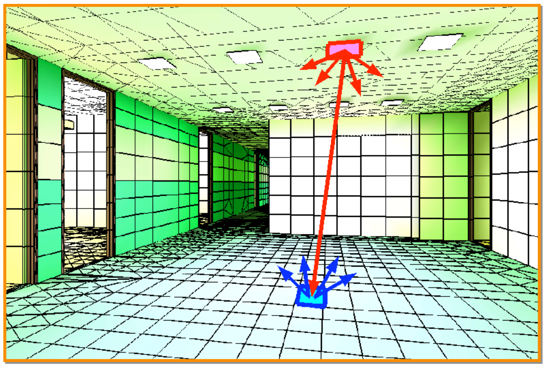
$$\begin{bmatrix} 1-\rho_1 F_{1,1} & . & . & . & -\rho_1 F_{1,n} \\ -\rho_2 F_{2,1} & 1-\rho_2 F_{2,2} & . & . & -\rho_2 F_{2,n} \\ . & . & . & . & . \\ . & . & . & . & . \\ -\rho_{n-1}F_{n-1,1} & . & . & . & -\rho_{n-1}F_{n-1,n} \\ -\rho_n F_{n,1} & . & . & . & 1-\rho_n F_{n,n} \end{bmatrix} \begin{bmatrix}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

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

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

← This is an energy balance equation

Radiosity Intuition



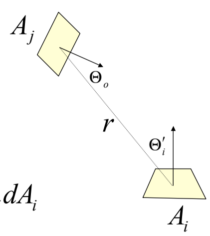
Radiosity

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

Form Factor

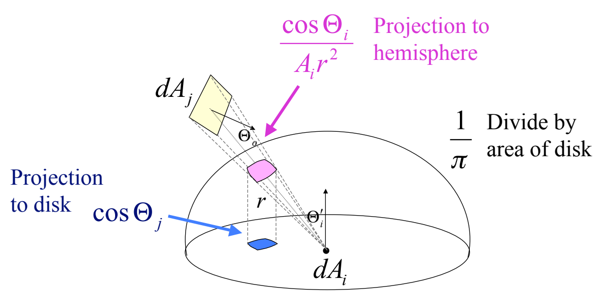
- Fraction of energy leaving element i that arrives at element j

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

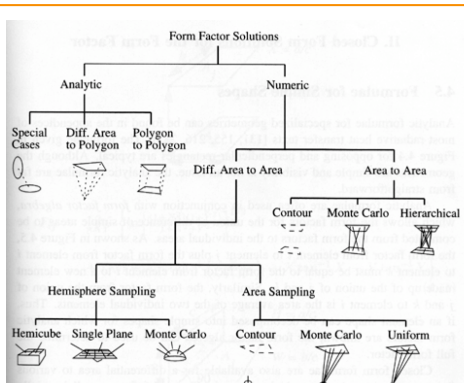


Form Factor Intuition

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



Computing Form Factors



Cohen & Wallace

Solving the System of Equations

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

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

A **x** = **b**

Solving the System of Equations

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

$$\underbrace{\begin{bmatrix} 1-\rho_1F_{1,1} & . & . & . & -\rho_1F_{1,n} \\ -\rho_2F_{2,1} & 1-\rho_2F_{2,2} & . & . & -\rho_2F_{2,n} \\ . & . & . & . & . \\ . & . & . & . & . \\ -\rho_{n-1}F_{n-1,1} & . & . & . & -\rho_{n-1}F_{n-1,n} \\ -\rho_nF_{n,1} & . & . & . & 1-\rho_nF_{n,n} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ . \\ B_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} E_1 \\ E_2 \\ . \\ . \\ . \\ E_n \end{bmatrix}}_{\mathbf{b}}$$

Gauss-Seidel Iteration

for all i
  $B_i = E_i$
while not converged
  for each i in turn
    $B_i = E_i + \rho_i \sum_{j \neq i} B_j F_{ij}$
display the image using B_i as the intensity of patch i.

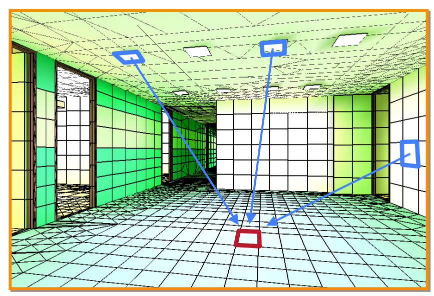
Gauss-Seidel Iteration

- Two interpretations:
  - Iteratively relax rows of linear system
  - Iteratively gather radiosity to elements

$$\begin{bmatrix} 1-\rho_1F_{1,1} & . & . & . & -\rho_1F_{1,n} \\ -\rho_2F_{2,1} & 1-\rho_2F_{2,2} & . & . & -\rho_2F_{2,n} \\ . & . & . & . & . \\ . & . & . & . & . \\ -\rho_{n-1}F_{n-1,1} & . & . & . & -\rho_{n-1}F_{n-1,n} \\ -\rho_nF_{n,1} & . & . & . & 1-\rho_nF_{n,n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ . \\ . \\ . \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ . \\ . \\ . \\ E_n \end{bmatrix}$$

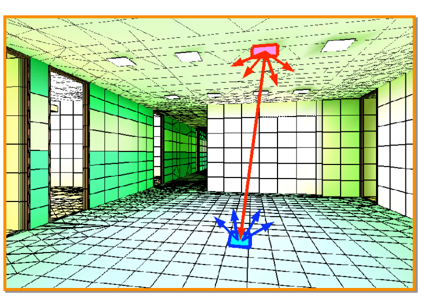
Gauss-Seidel Iteration

- Two interpretations:
  - Iteratively relax rows of linear system
  - Iteratively gather radiosity to elements



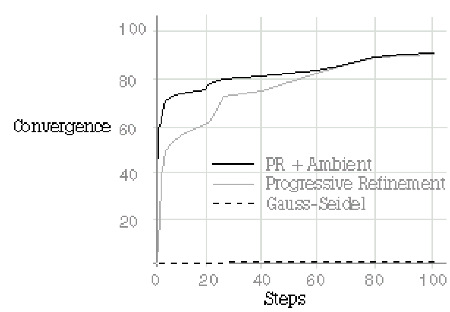
Progressive Radiosity

- Interpretation:
  - Iteratively shoot “unshot” radiosity from elements
  - Select shooters in order of unshot radiosity



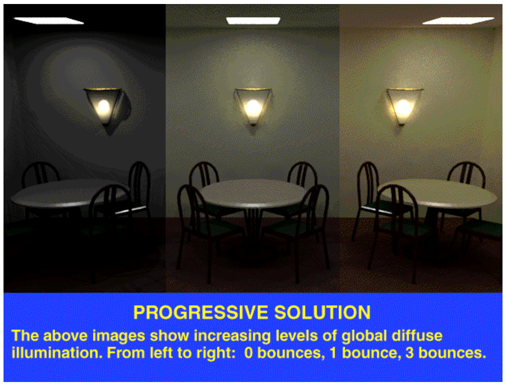
Progressive Radiosity

- Adaptive refinement

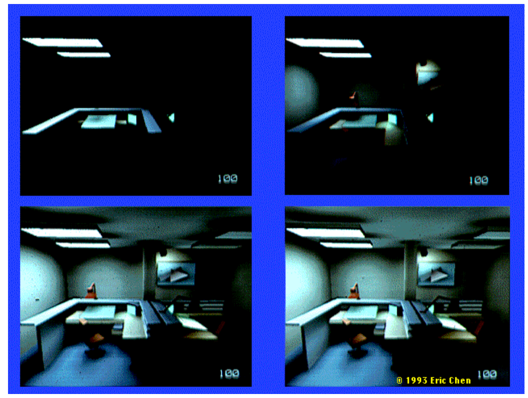


Yeap

Progressive Radiosity



Progressive Radiosity



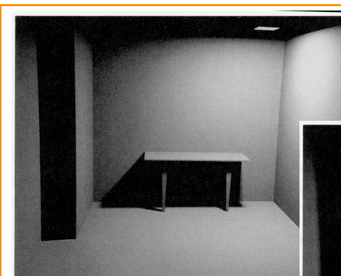
Surface Meshing

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

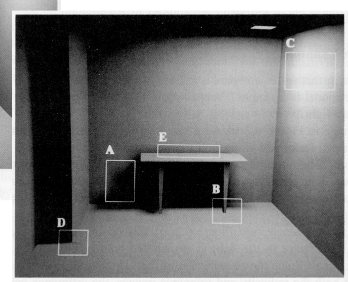
Figure 6.2: *A radiosity image computed using a uniform mesh.*

Cohen & Wallace

Artifacts of Bad Surface Meshing



(a) *Reference image.*

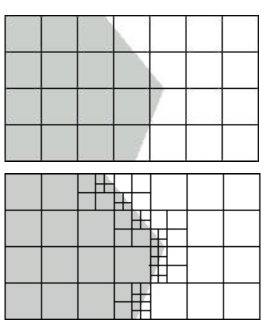


(b) *Artifacts introduced by the approximation.*

Cohen & Wallace

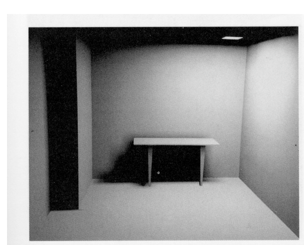
Adaptive Meshing

- Refine mesh in areas of high residual

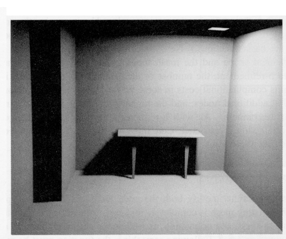


Yeap

Adaptive Meshing



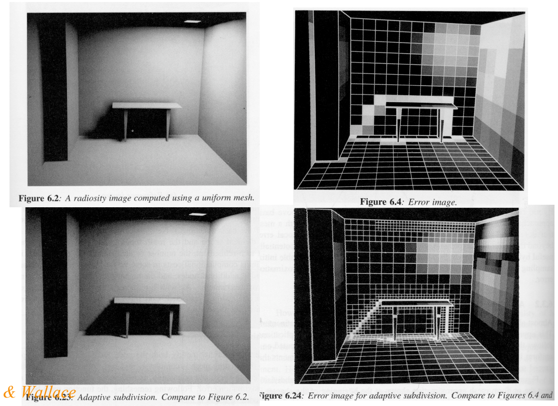
Uniform mesh



Adaptive mesh

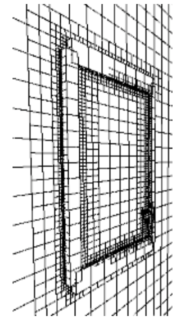
Cohen & Wallace

Error Comparison

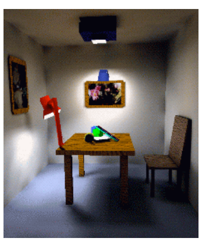


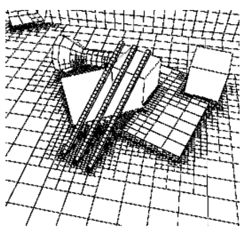
Cohen & Wallace

Adaptive Meshing



Adaptive Meshing





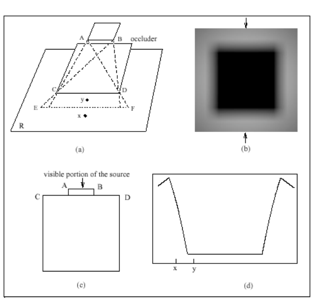
(table top from different angle)

Adaptive Meshing

Baum et al.

Discontinuity Meshing

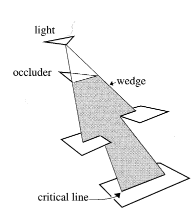
- Capture discontinuities in radiosity across a surface with explicit mesh boundaries



Lischinski et al.

Discontinuity Meshing

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



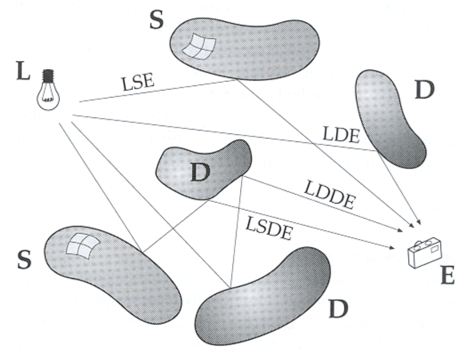
Discontinuity
Mesh

Lischinski et al.

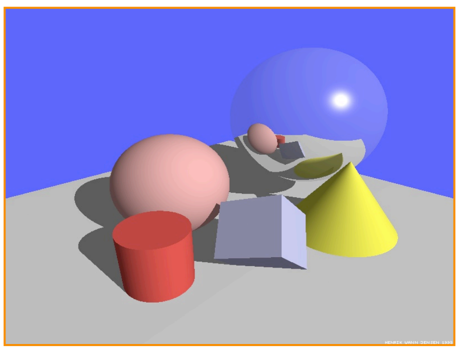
Overview

- Global illumination
  - o Rendering equation
- Solution methods
  - o OpenGL
  - o Ray tracing
  - o Path tracing
  - o Radiosity
- Path types
  - o L(S|D)*E

Path Types



Path Types?

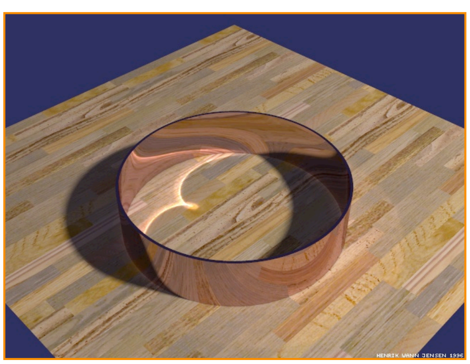


Henrik Wann Jensen

Path Types?

Paul Debevec

Path Types?



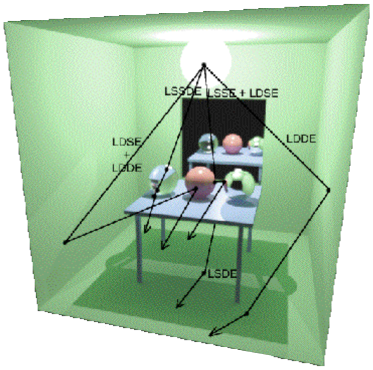
Henrik Wann Jensen

Path Types?

RenderPark

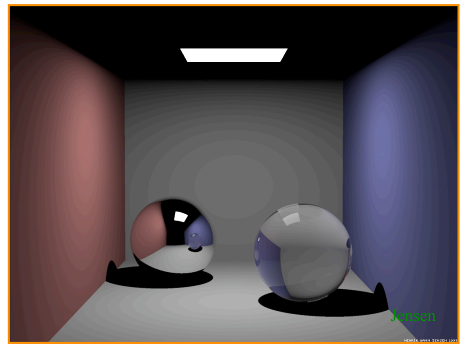
Path Types

- OpenGL
  - o LDE
- Ray tracing
  - o LDS*E
- Radiosity
  - o LD*E
- Path tracing
  - o L(DIS)*E



John Hart

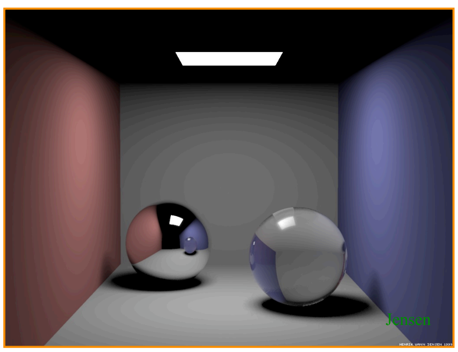
Path Types



Ray tracing

Henrik Wann Jensen

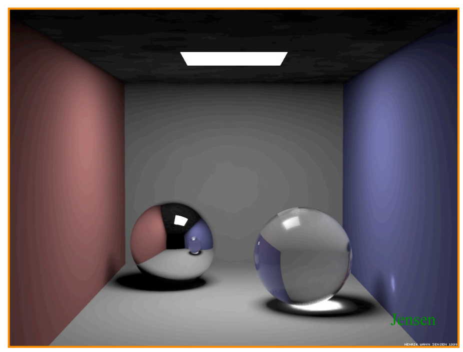
Path Types



+ soft shadows

Henrik Wann Jensen

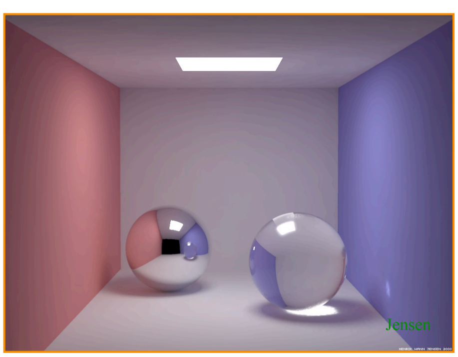
Path Types



+ caustics

Henrik Wann Jensen

Path Types



+ indirect diffuse illumination

Henrik Wann Jensen

Summary

- Global illumination
  - o Rendering equation
- Solution methods
  - o OpenGL
  - o Ray tracing
  - o Radiosity
  - o Path tracing
- Path types
  - o L(SID)*E