Radiosity

Doug James

Carnegie Mellon University

15-864 Advanced Computer Graphics

Acknowledgements

- Tom Funkhouser
- Dutré et al. 2003, Chapter 6
- Keller, “Instant Radiosity,” SIGGRAPH 97.

Radiosity

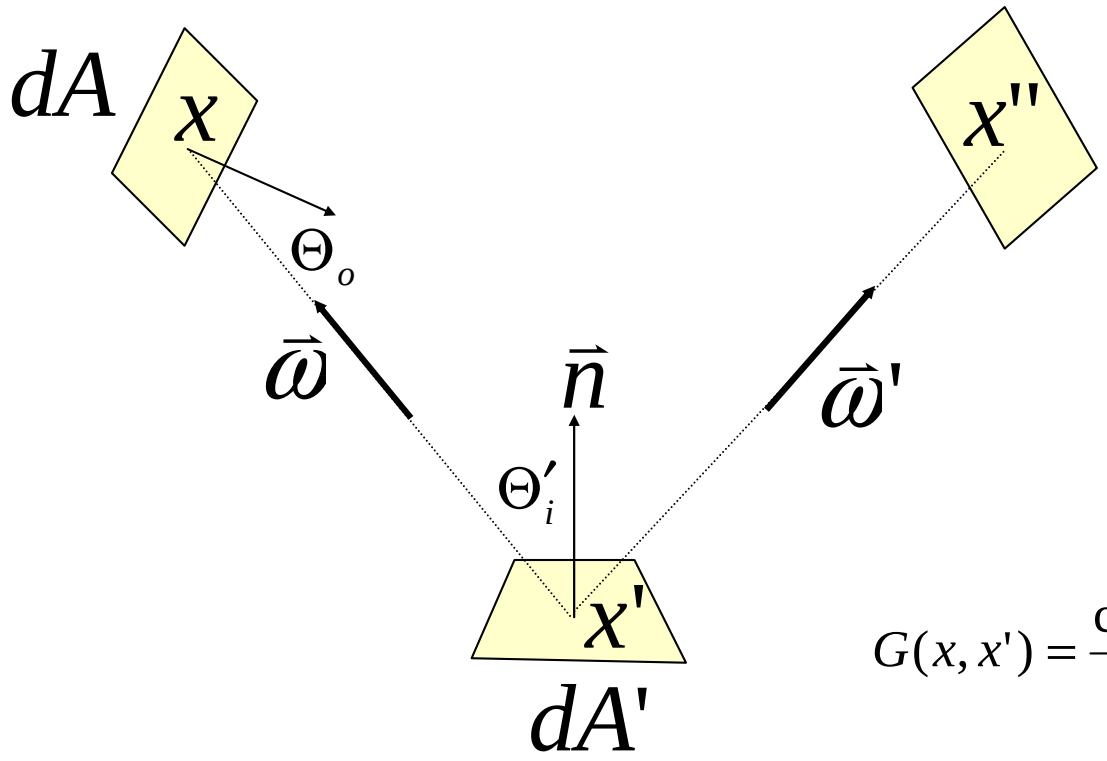

John R. Wallace
© 1988 Program of Computer Graphics
Cornell University

Overview

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

Rendering Equation

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



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

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

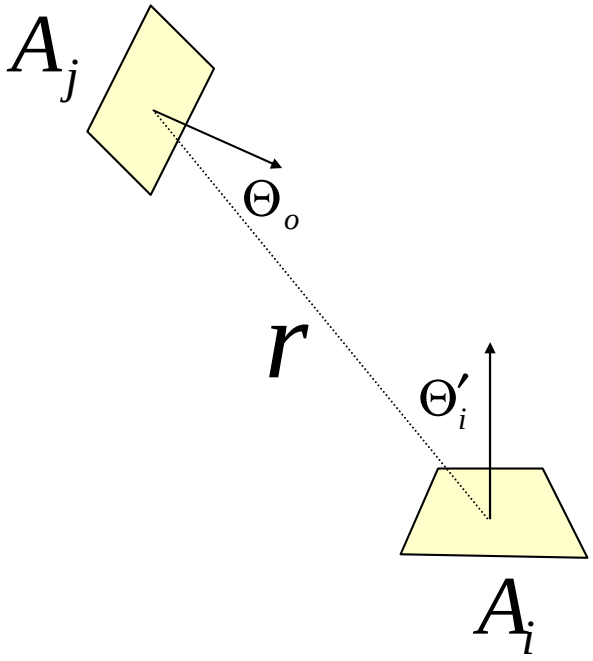
$$B = \int_\Omega L_o \cos\theta d\omega \qquad L = \frac{B}{\pi}$$

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

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

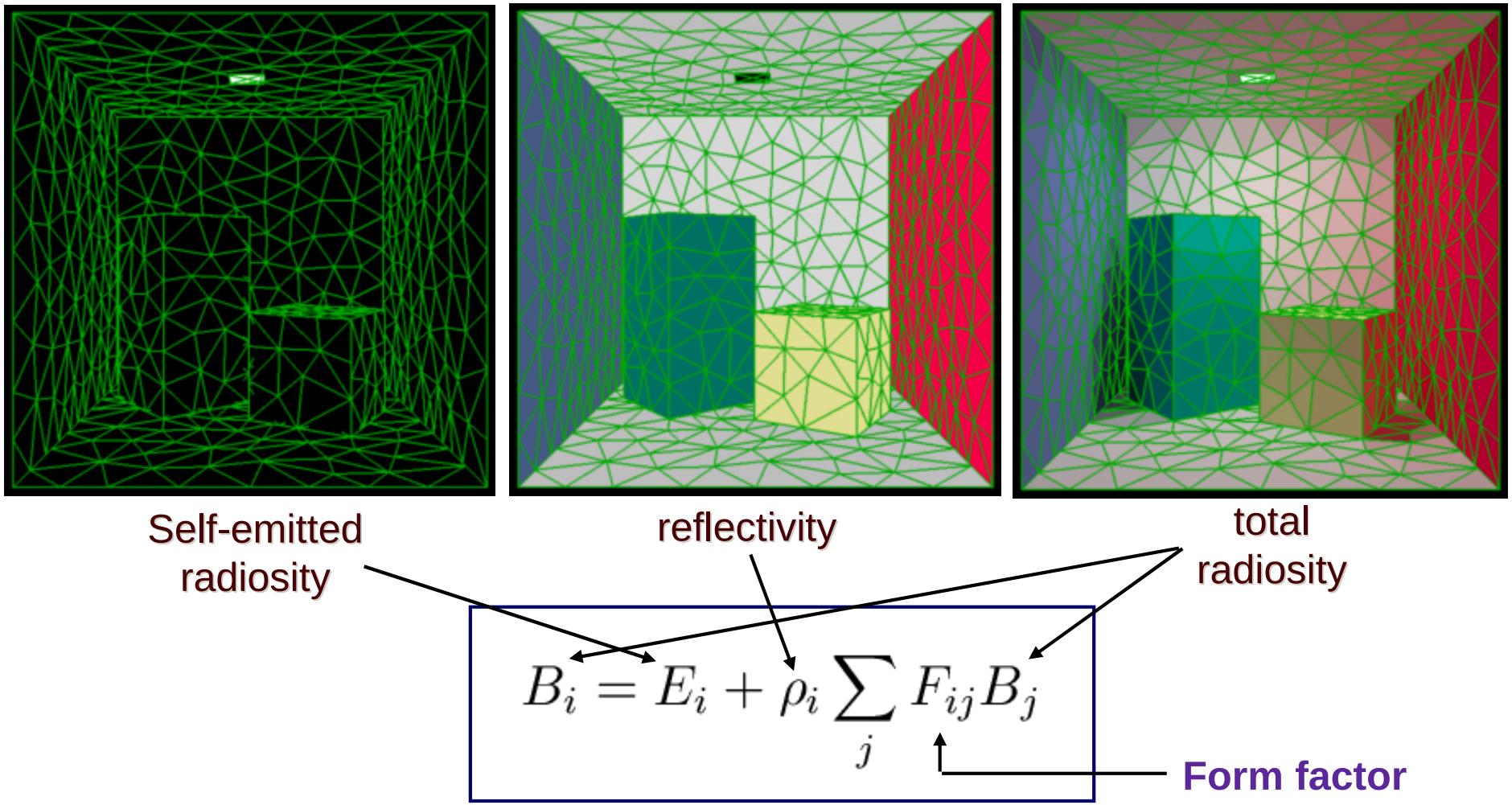


Fig 6.1, [Dutré et al. 2003]

System of Equations

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

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

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

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

Intuition

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

Multiply by A_i

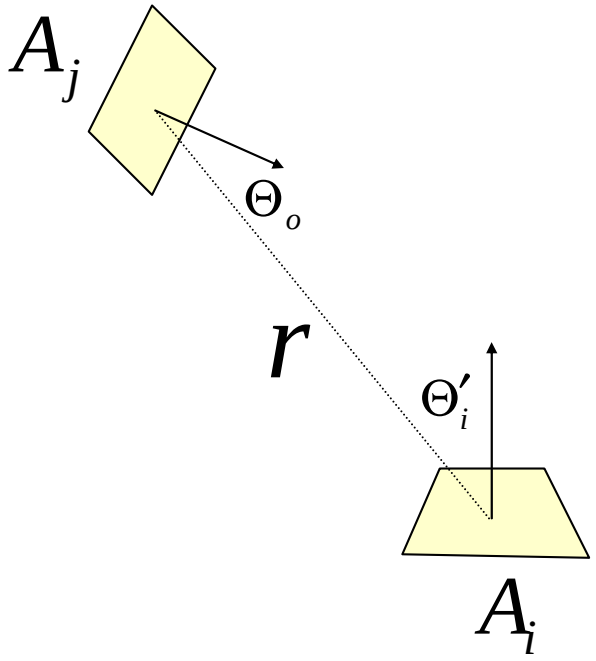
$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^{N} B_j F_{ij} A_i$$

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

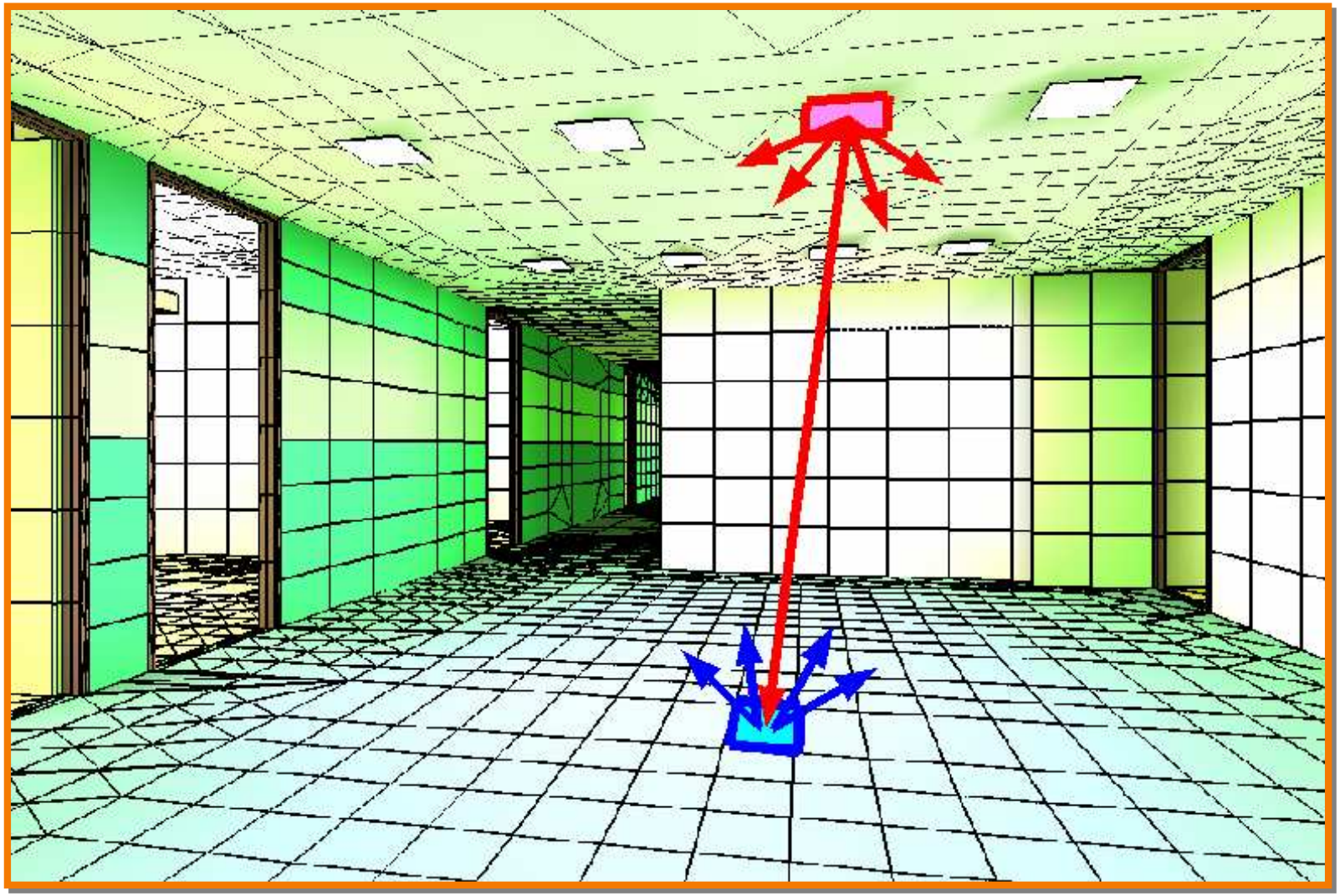
$$B_i A_i = E_i A_i + \rho_i \sum_{j=1}^{N} B_j F_{ji} A_j$$

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

← This is an energy balance equation



Intuition

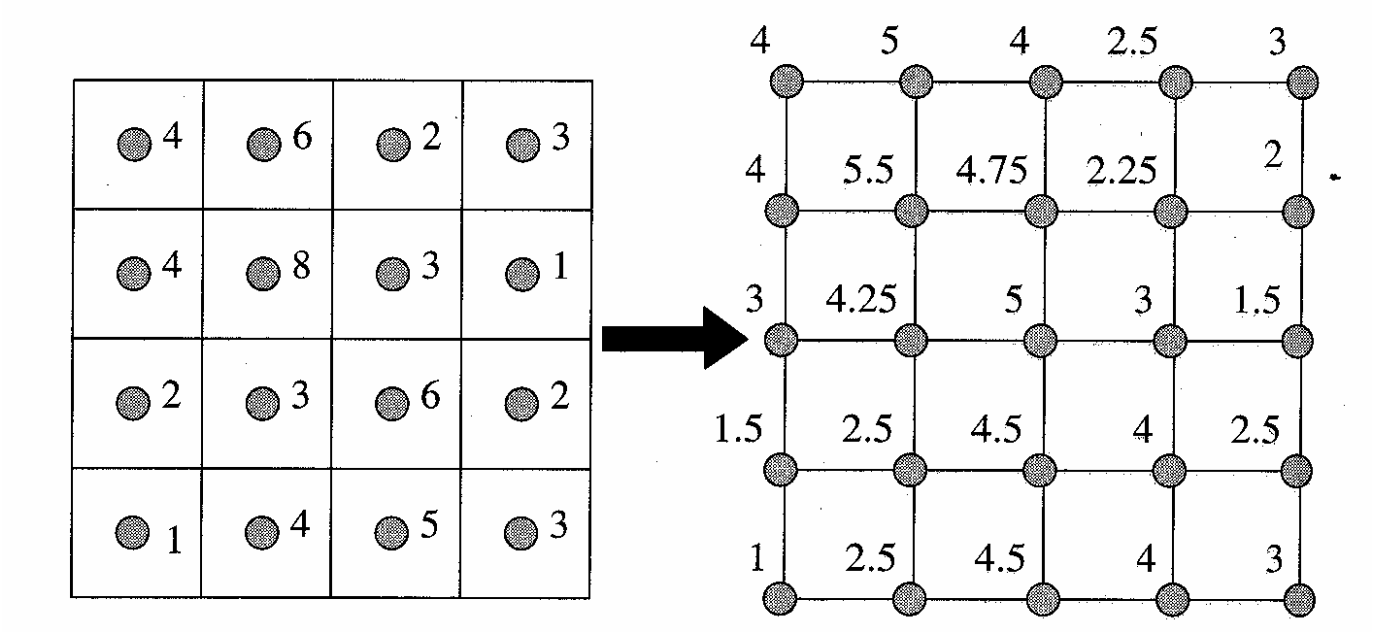


Overview

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

Displaying Radiosity

- Usually, Gouraud shading …



computed displayed

Discretisation Artifacts

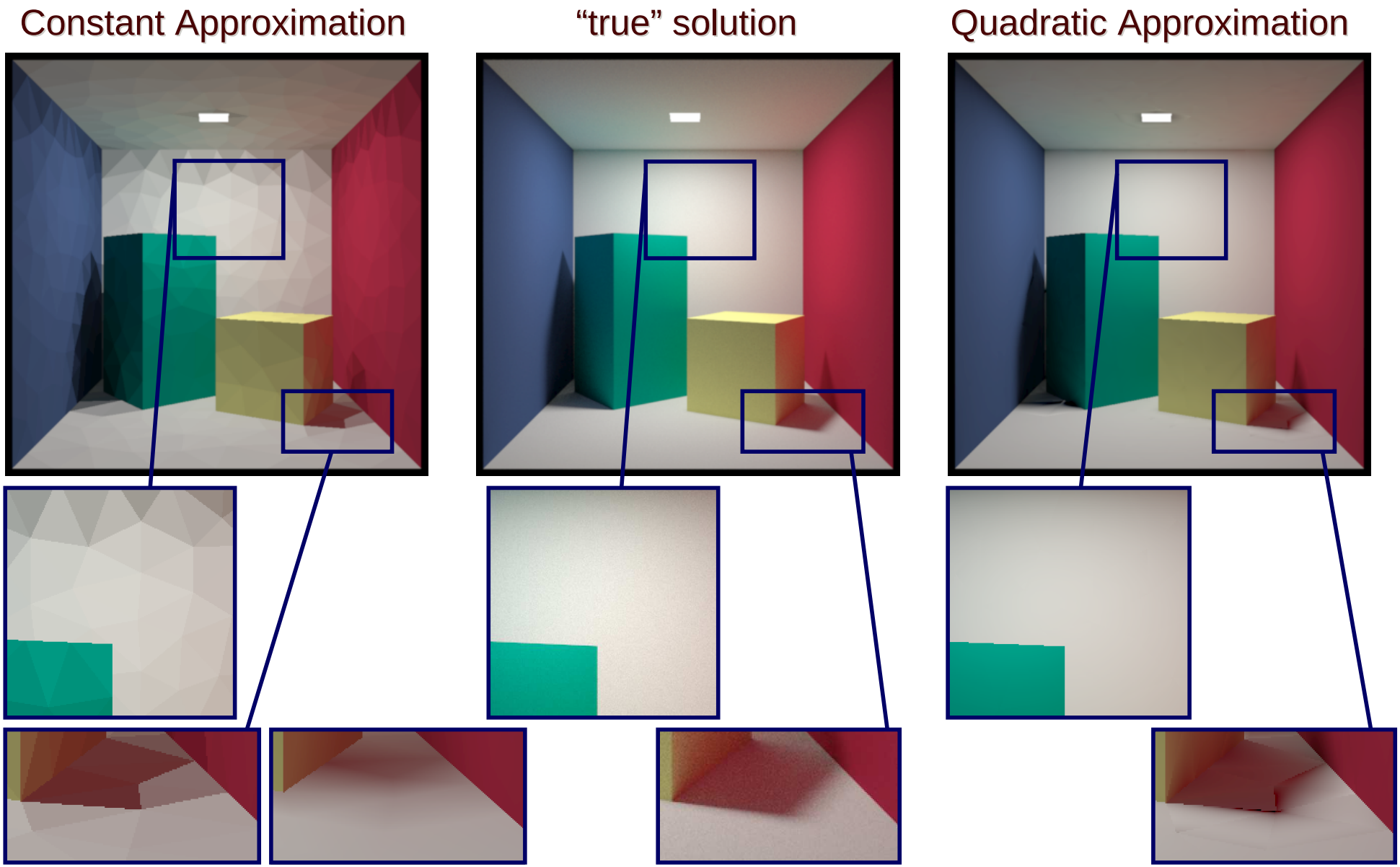


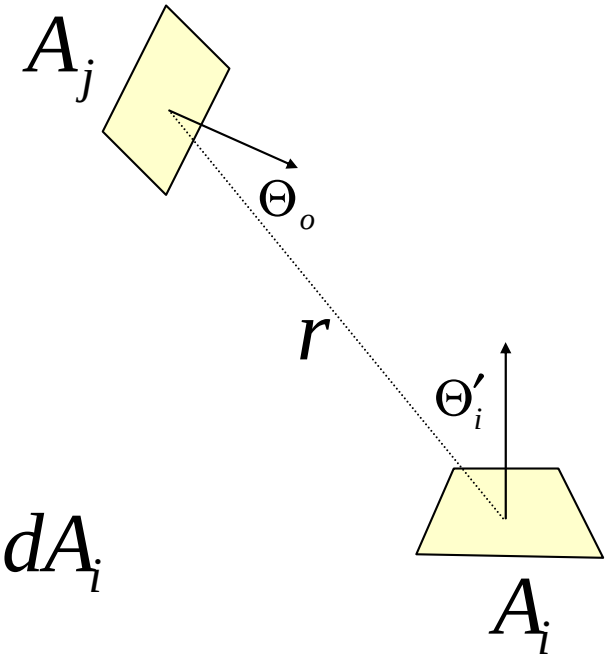
Fig 6.2, [Dutré et al. 2003]

Overview

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

Form Factor

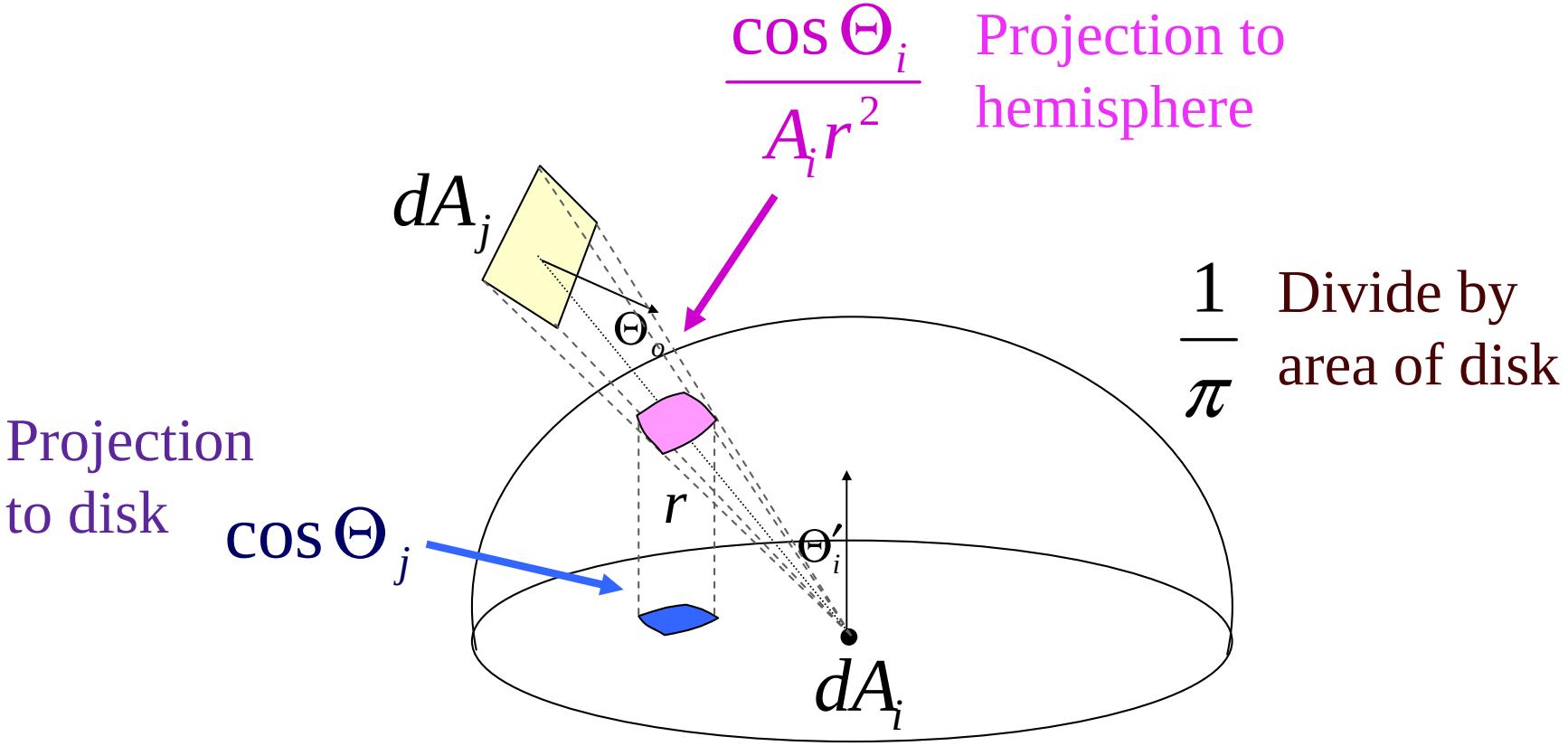
- Fraction of energy leaving element i that arrives at element j



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

Form Factor Intuition

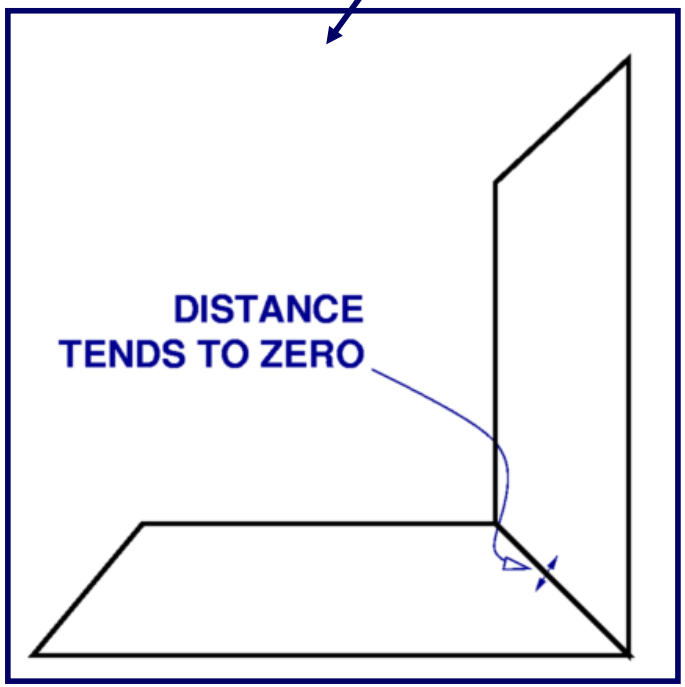
$$F_{di-dj} = \frac{1}{A_i} \frac{V_{ij} \cos\Theta_i \cos\Theta_j}{\pi r^2}$$



Form Factor Singularities and Discontinuities

$$F_{ij} = \frac{1}{A_i} \int_{S_i} \int_{S_j} G(x,y)\, dA_y dA_x$$

$$G(x,y) = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} \mathrm{vis}(x,y).$$



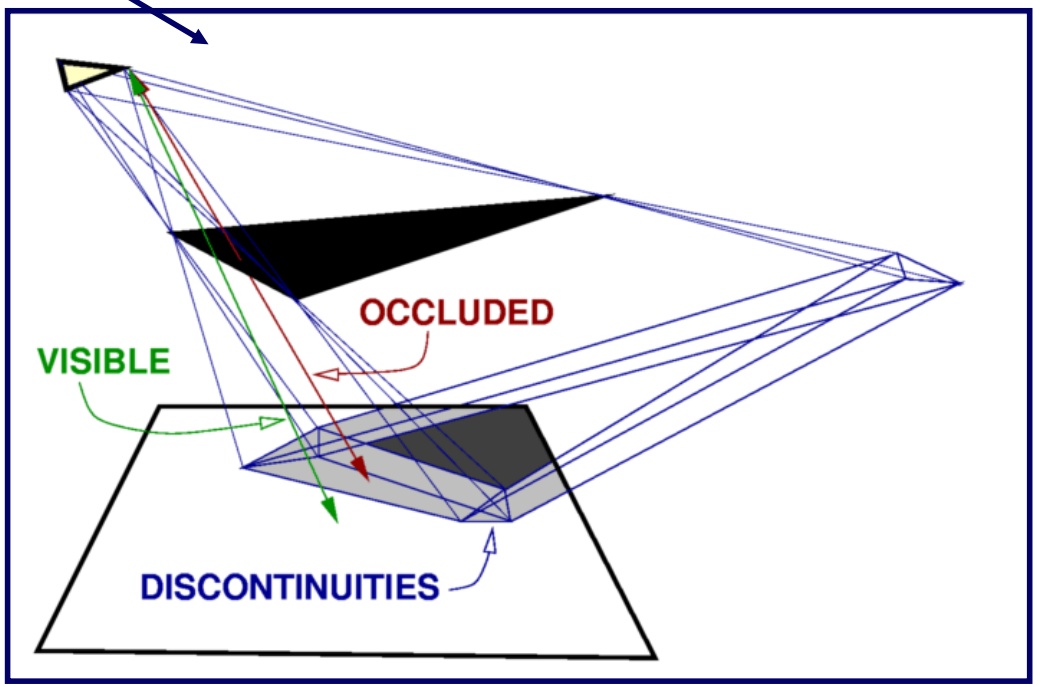
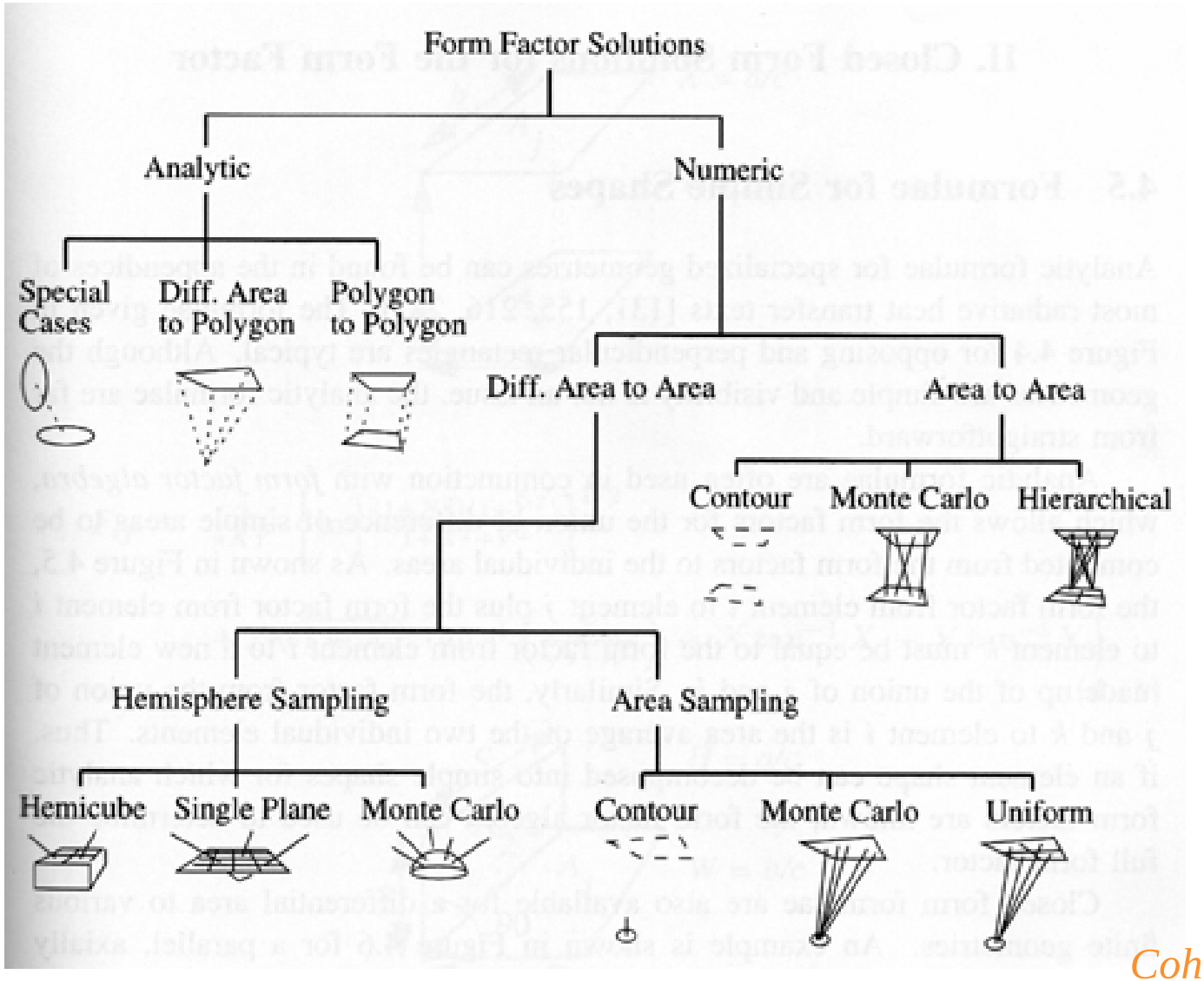


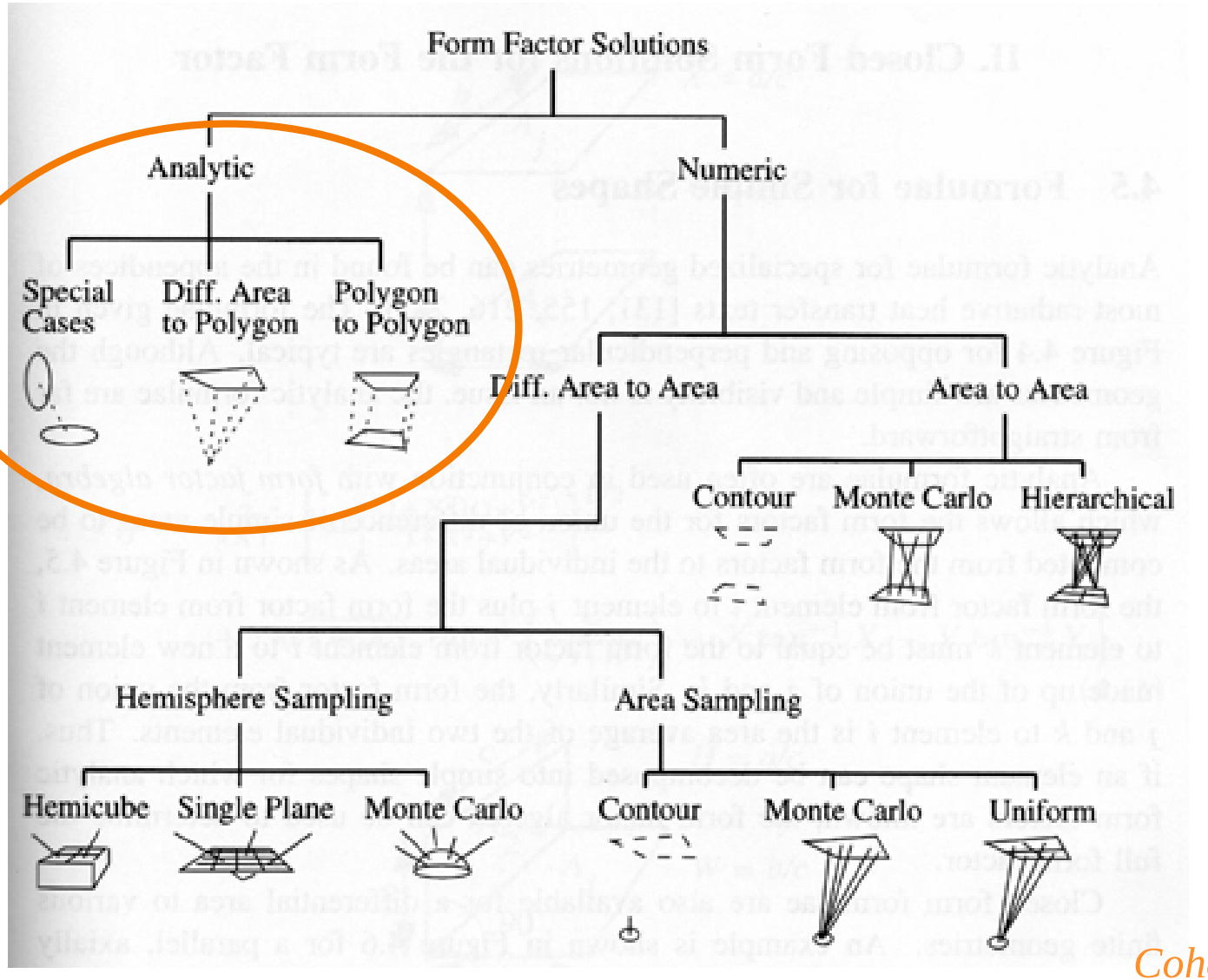
Fig 6.3, [Dutré et al. 2003]

Computing Form Factors



Cohen & Wallace

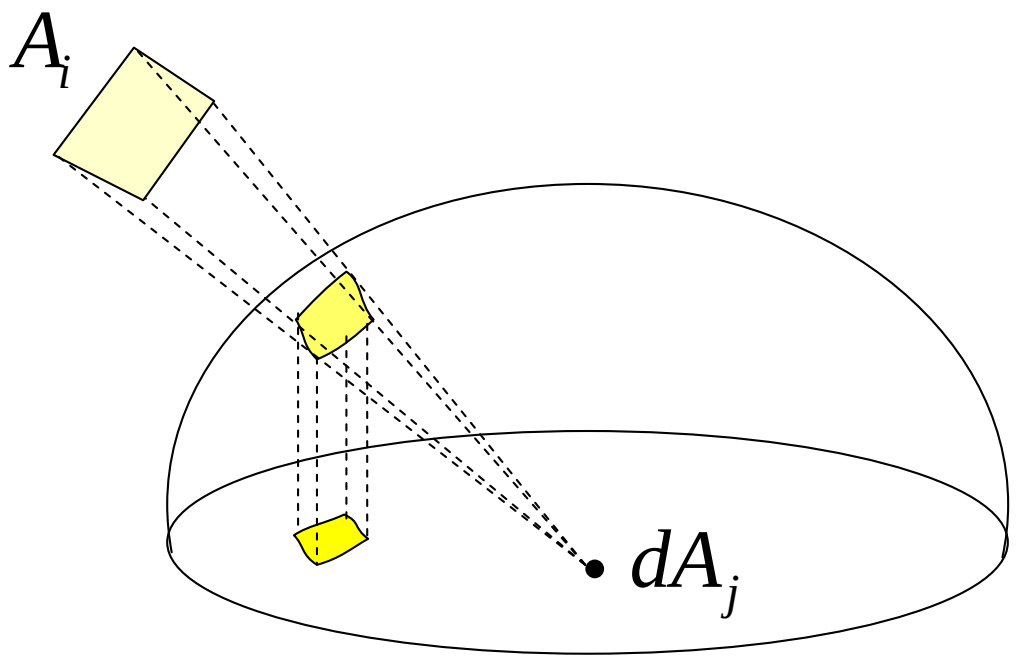
Computing Form Factors



Cohen & Wallace

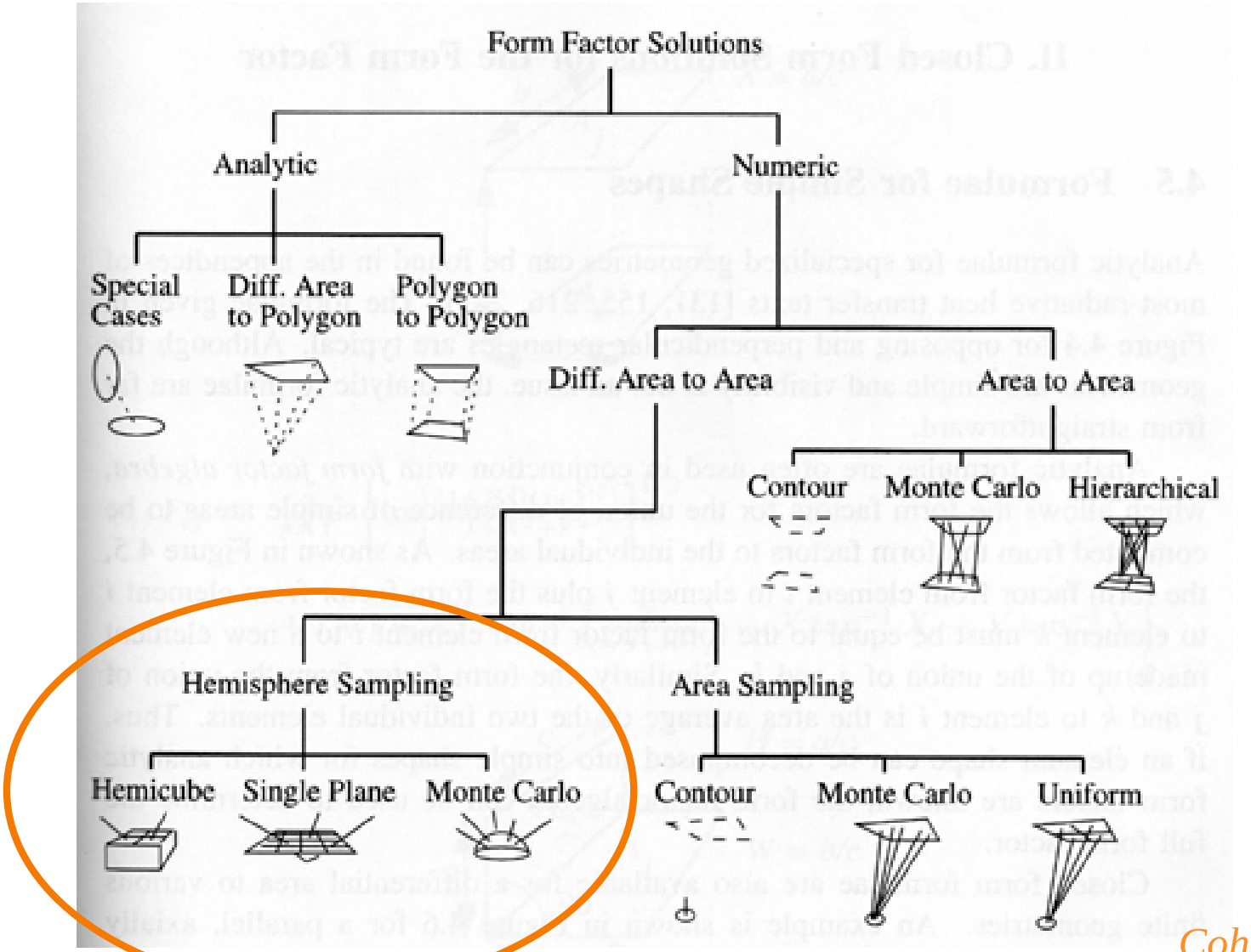
Analytic Form Factors

- Derive equation for projected area
  - Possible only for simple situations
    e.g., direct illumination of a point from a visible polygon



Partial visibility is problematic

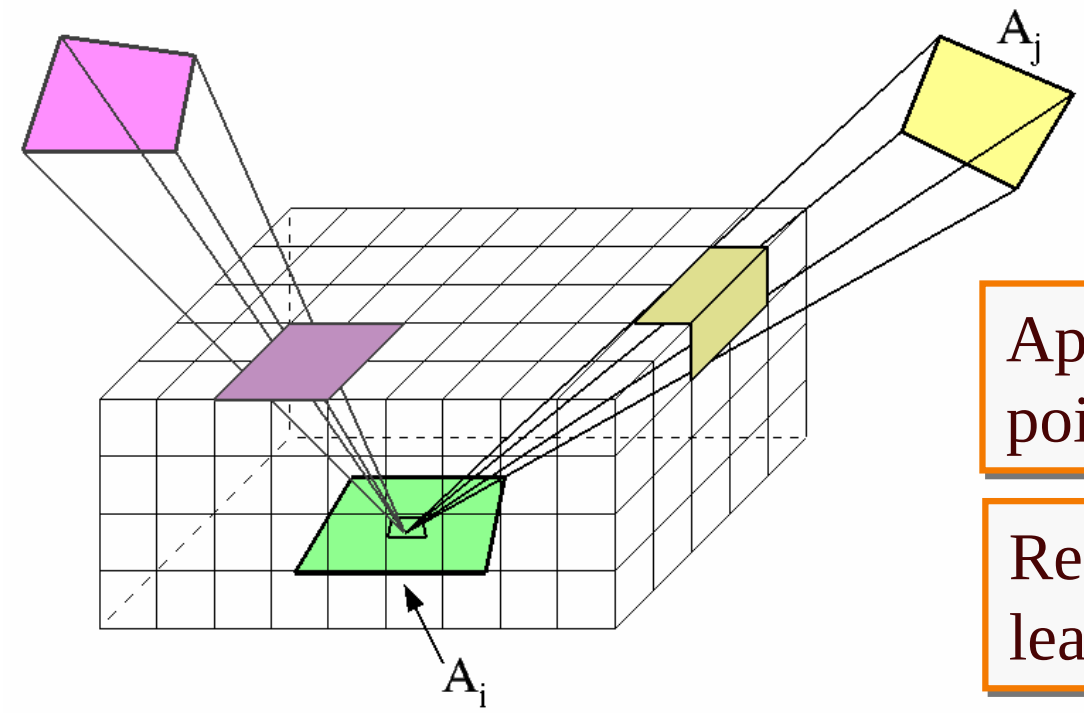
Computing Form Factors



Cohen & Wallace

Hemicube [Cohen & Greenberg, 85]

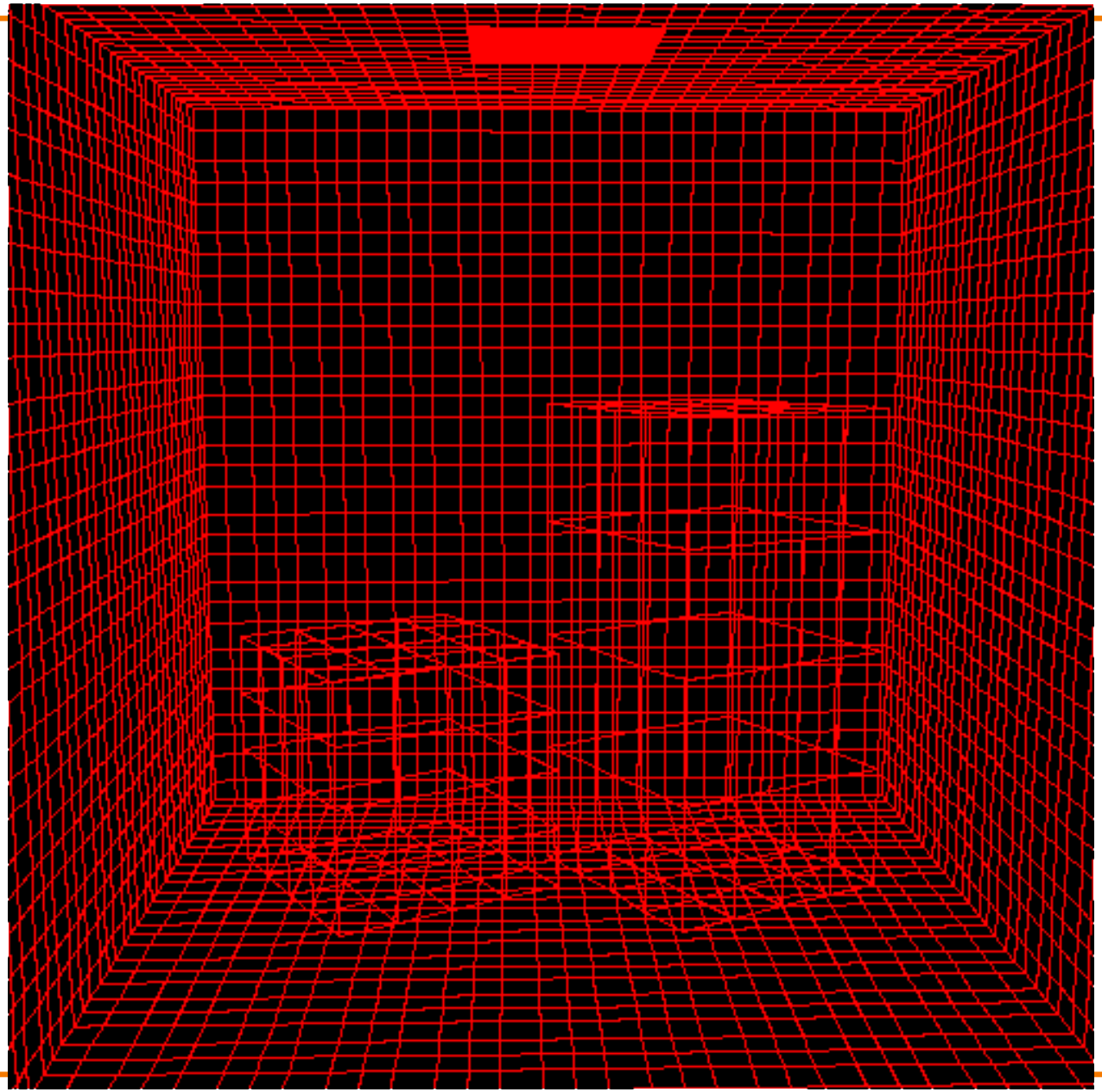
- Compute form factor with image-space precision
  - Render scene from centroid of A_i
  - Use z-buffer to determine visibility of other surfaces
  - Count “pixels” to determine projected areas



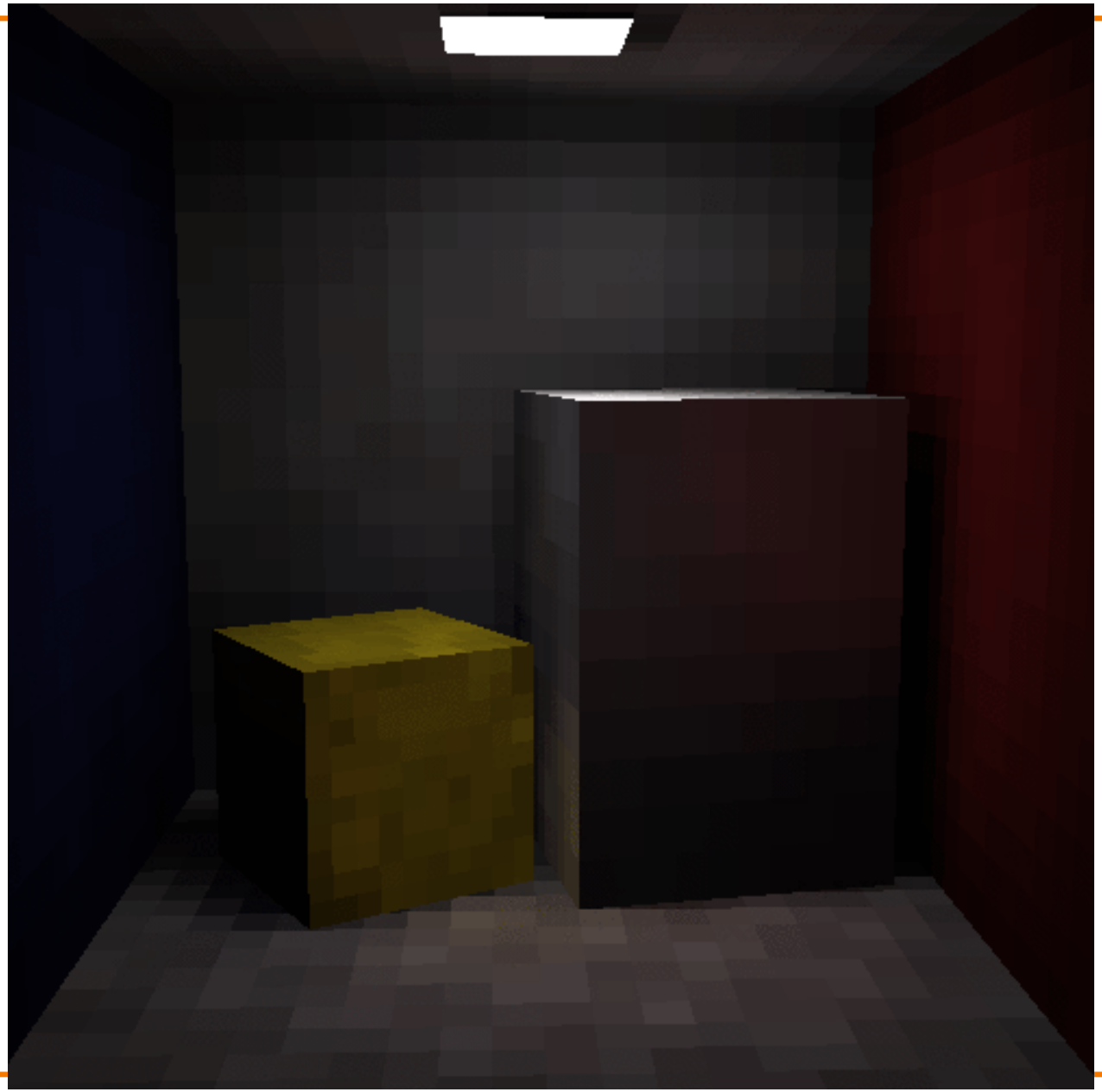
Approximating A_i with point leads to errors

Regular sampling leads to aliasing artifacts

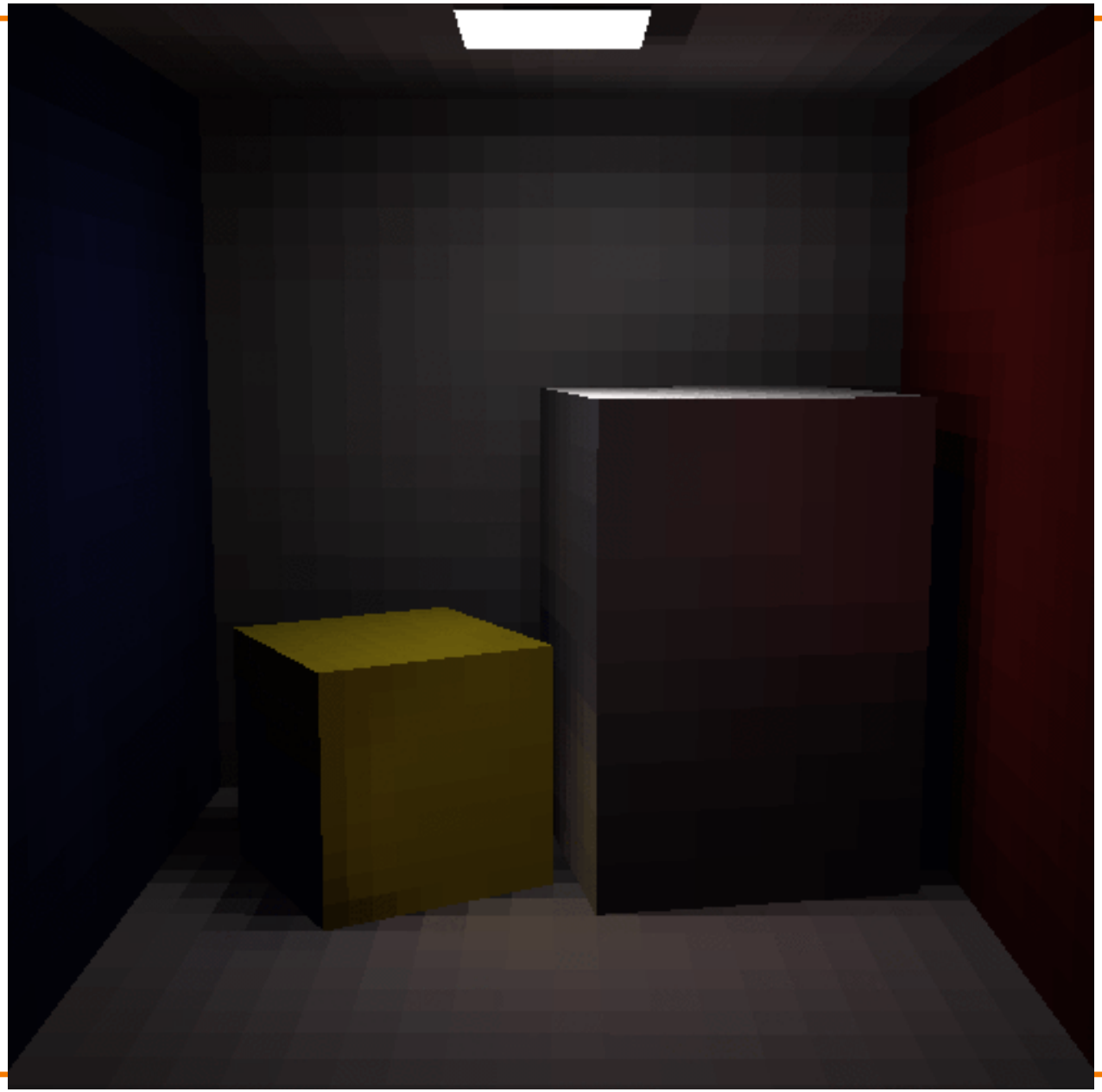
Ex: Wireframe



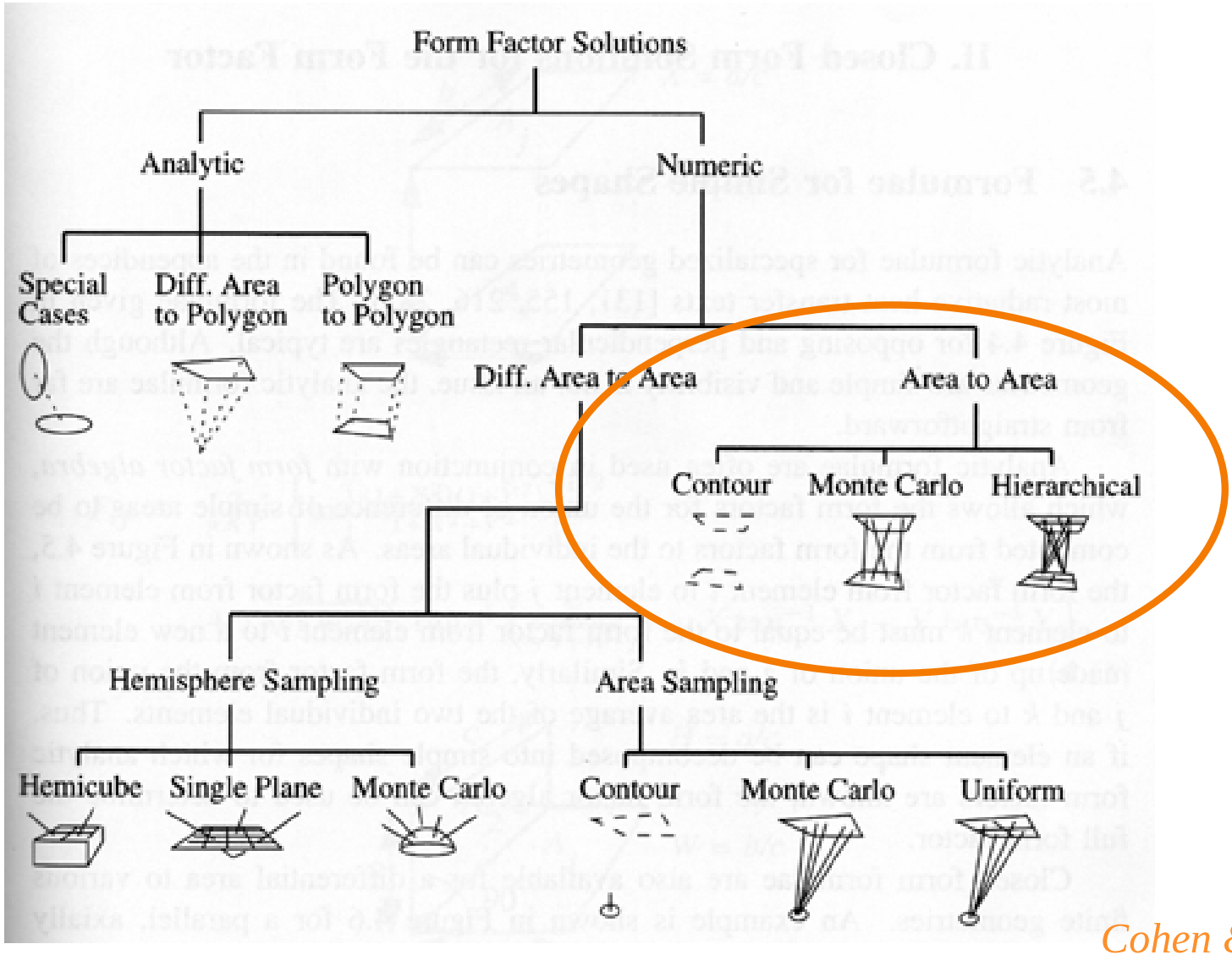
Ex: Classical HC (no interpolation)



Ex: Antialiased HC e.g. [Chandran et al. 2000]



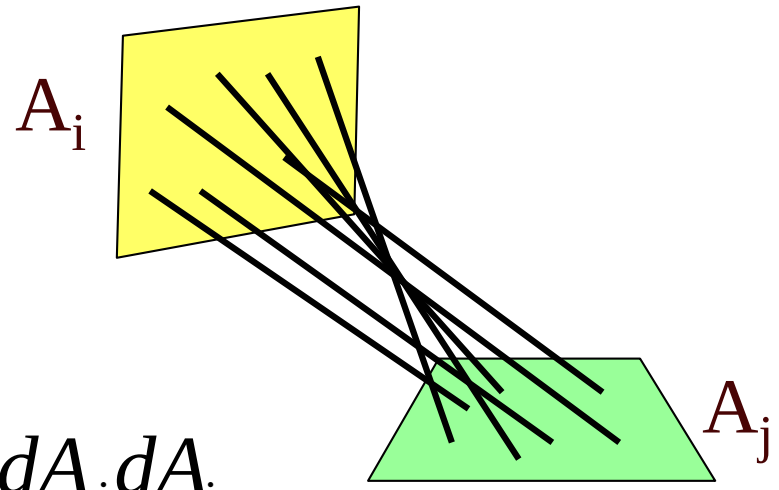
Computing Form Factors



Cohen & Wallace

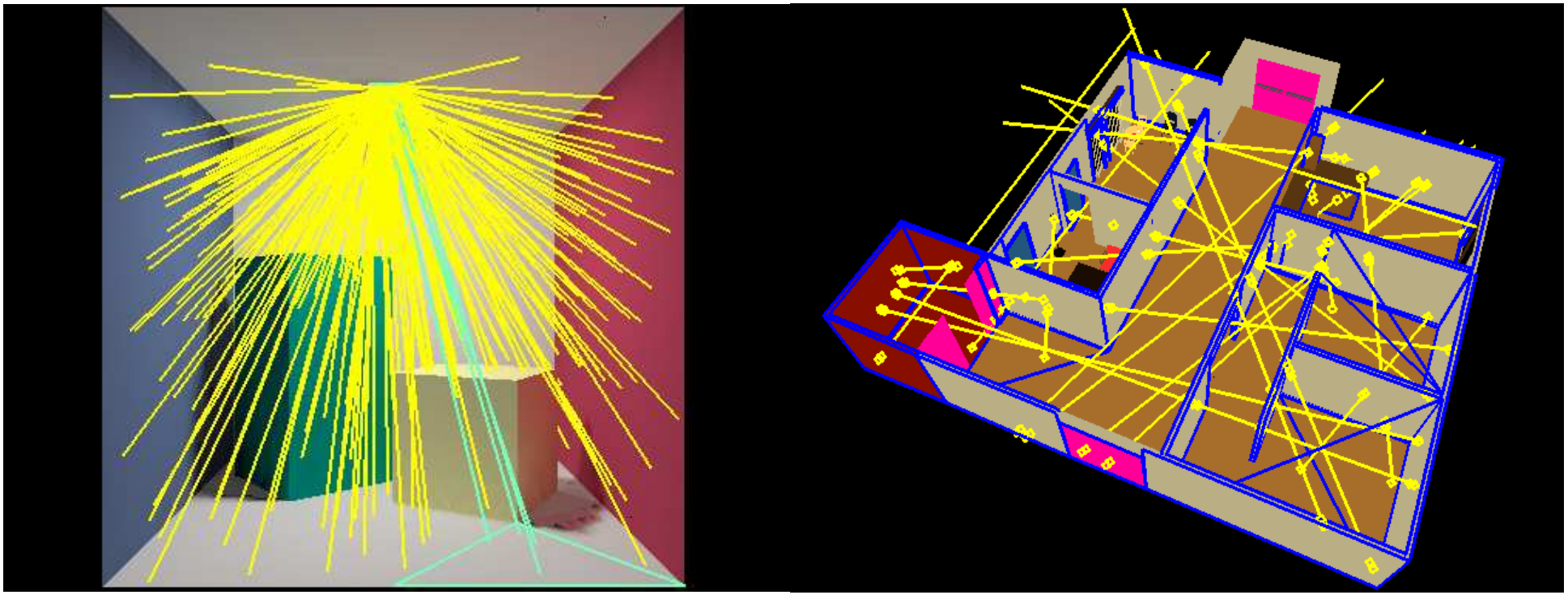
Monte Carlo Sampling

- 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



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

Other approaches: Global Lines



Local Lines

Global Lines

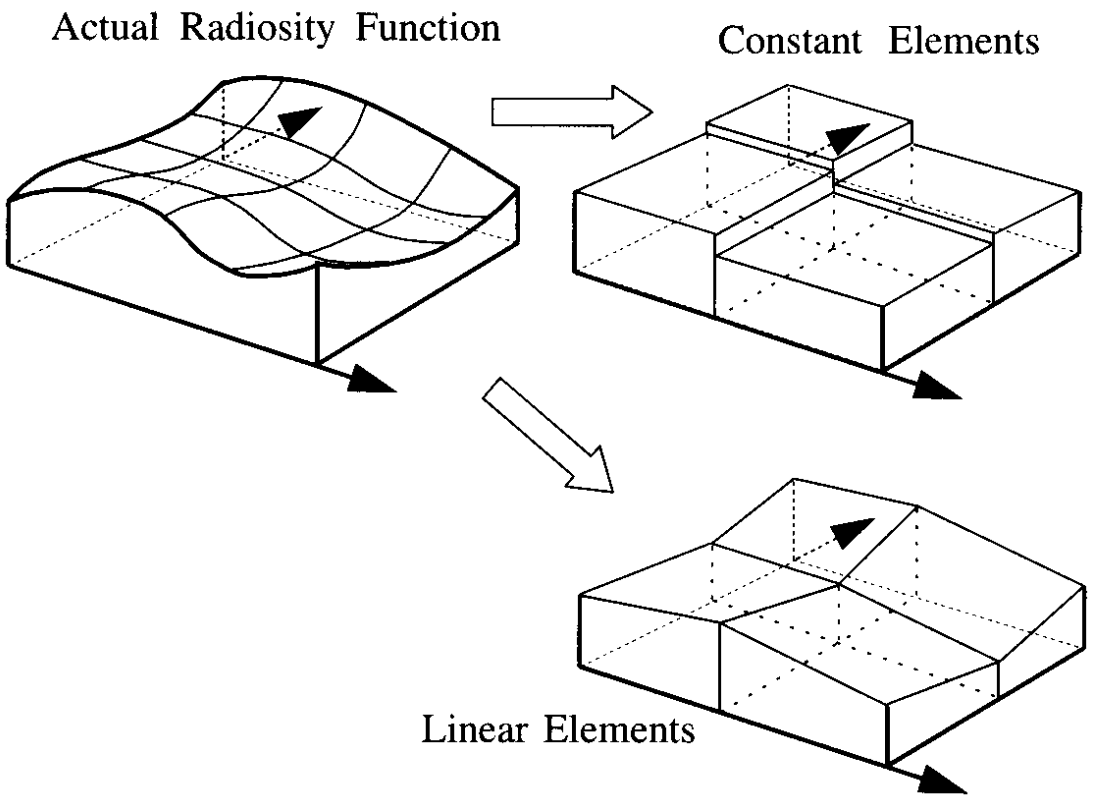
Fig 6.4, [Dutré et al. 2003]

Overview

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

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
- Discussion

Solving the System of Equations

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

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

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

Solving the System of Equations

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

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

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

Gauss-Seidel Iteration

1 for all i
2 $\quad B_i = E_i$
3 while not converged
4 $\quad$ for each i in turn
5 $\quad\quad B_i = E_i + \rho_i \sum_{j \neq i} B_j F_{ij}$
6 display the image using B_i as the intensity of patch i.

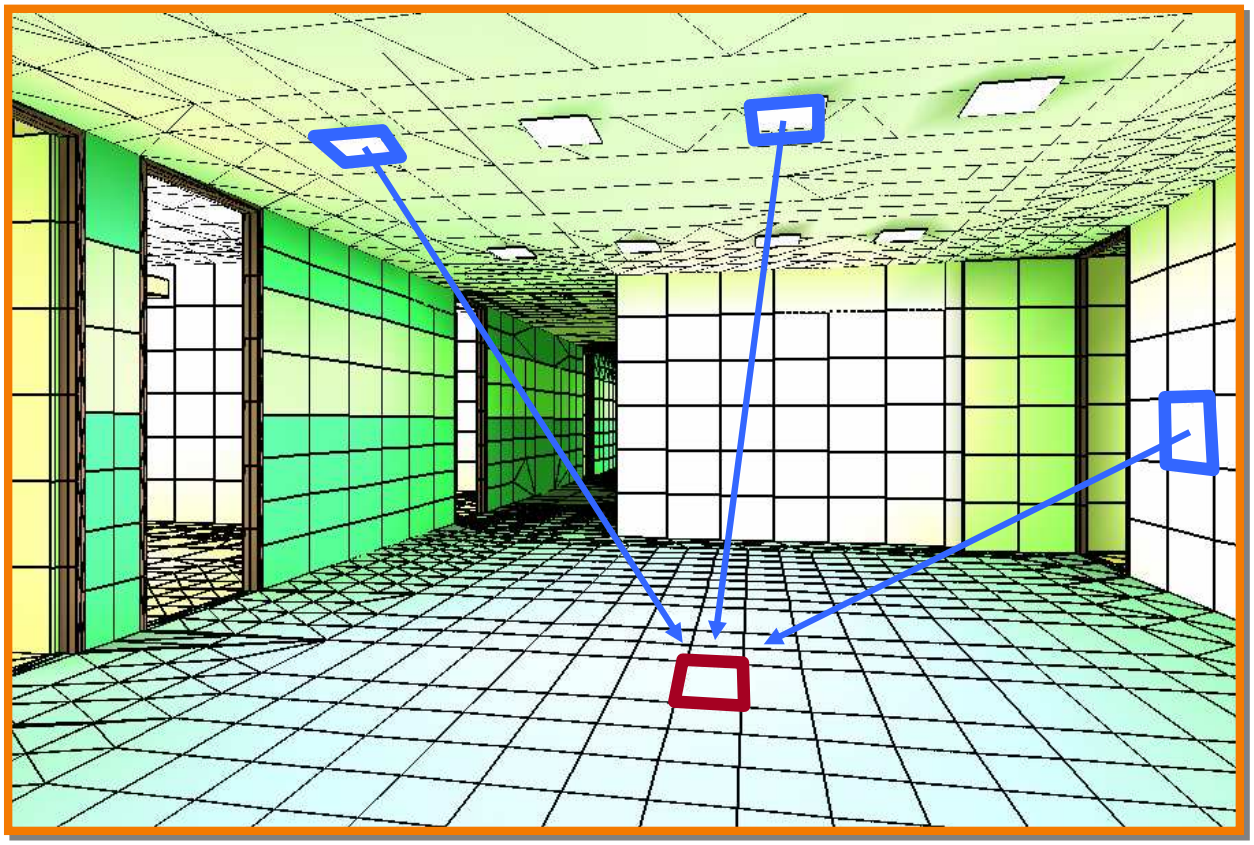
Gauss-Seidel Iteration

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

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

Gauss-Seidel Iteration

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



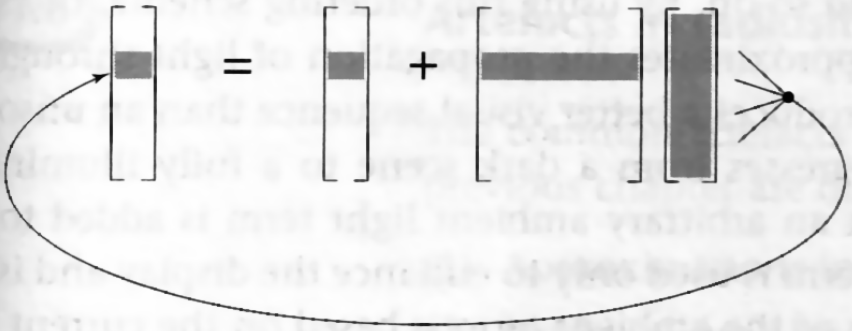
Progressive Radiosity

- Cohen et al., SIGGRAPH 1988
- Shoot light instead of gathering light
- Each iteration is O(n)
- May or may not keep $F_{i\,j}$ after each iteration

Gathering & Shooting Radiosity

Gathering: a single iteration (k) updates a single patch i by gathering contributions from all other patches.

$$B_i^{(k+1)} = E_i + R_i \sum_{j=1}^{N} F_{ij} B_j^{(k)}$$

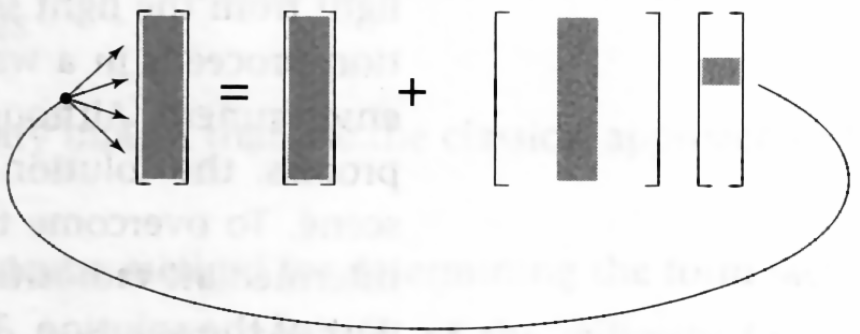


Equivalent to gathering light energy from all the patches in the scene.

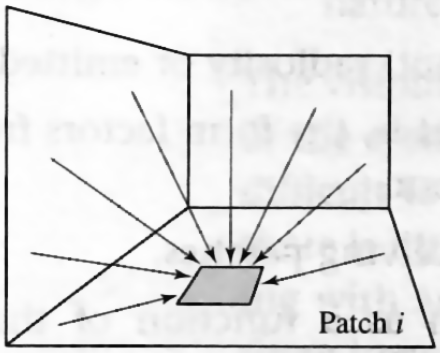
Shooting: a single step computes form factors from the shooting patch to all receiving patches and distributes (unshot) energy ΔB_i

for all j:

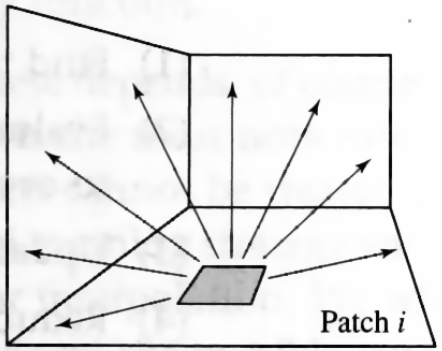
$$B_j^{(k+1)} = B_j^{(k)} + R_j F_{ji} \Delta B_i$$



Equivalent to shooting light energy from a patch to all other patches in the scene.



(a) Gathering

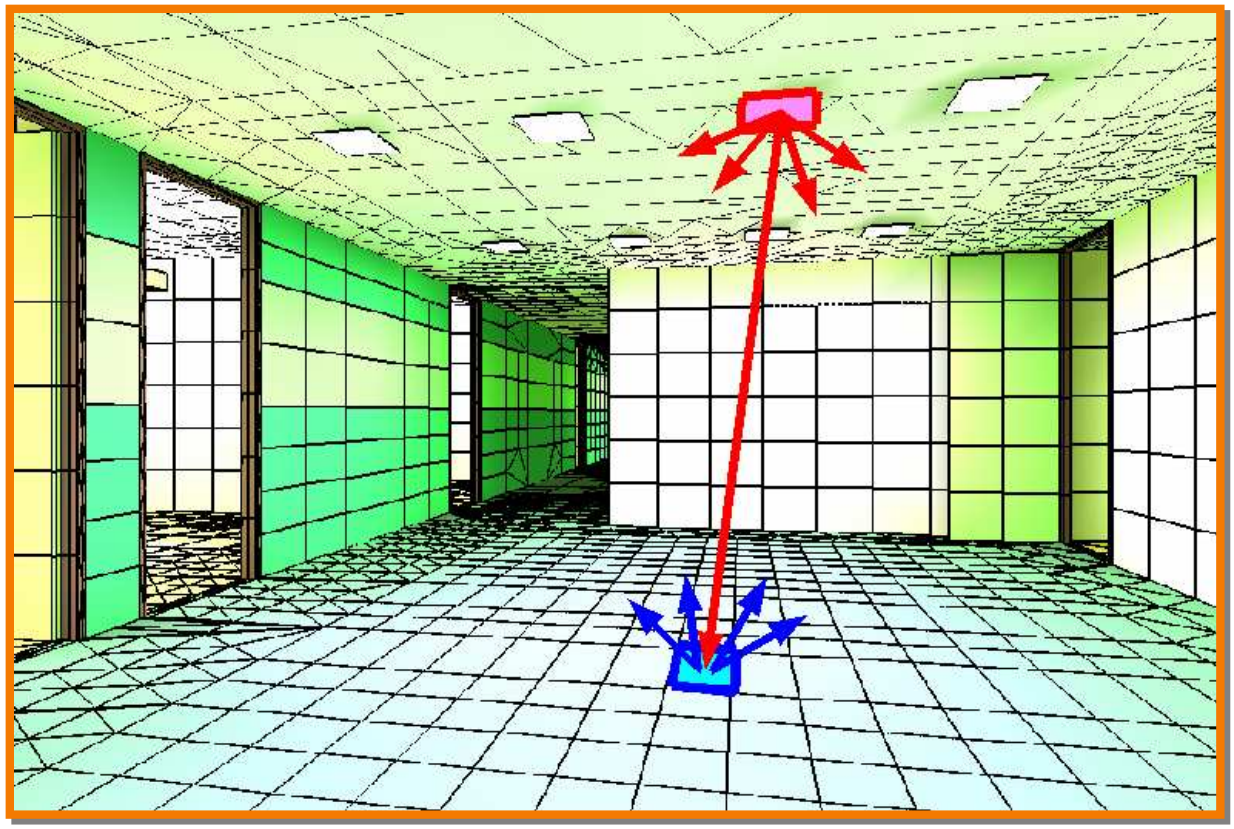


(b) Shooting

Watt, 3D CG, 3rd Ed, 2000

Progressive Radiosity

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



Progressive Radiosity

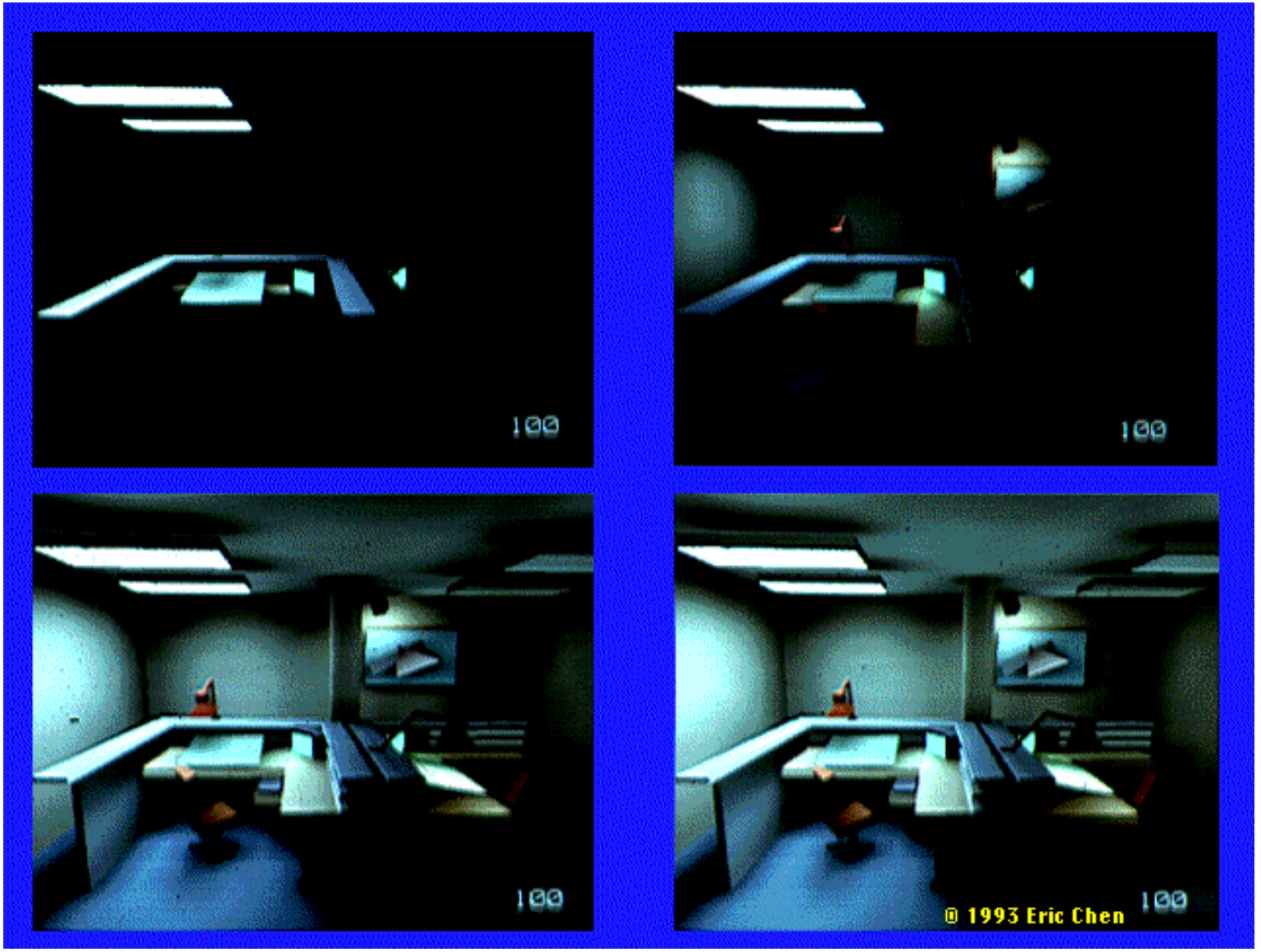
1 for all i
2 $B_i = E_i$
3 $\Delta B_i = E_i$
4 while not converged
5 pick i, such that $\Delta B_i * A_i$ is largest
6 for every patch j
7 $\Delta rad = \Delta B_i * \rho_j F_{ji}$
8 $\Delta B_j = \Delta B_j + \Delta rad$
9 $B_j = B_j + \Delta rad$
10 $\Delta B_i = 0$
11 display the image using B_i as the intensity of patch i.

Progressive Radiosity

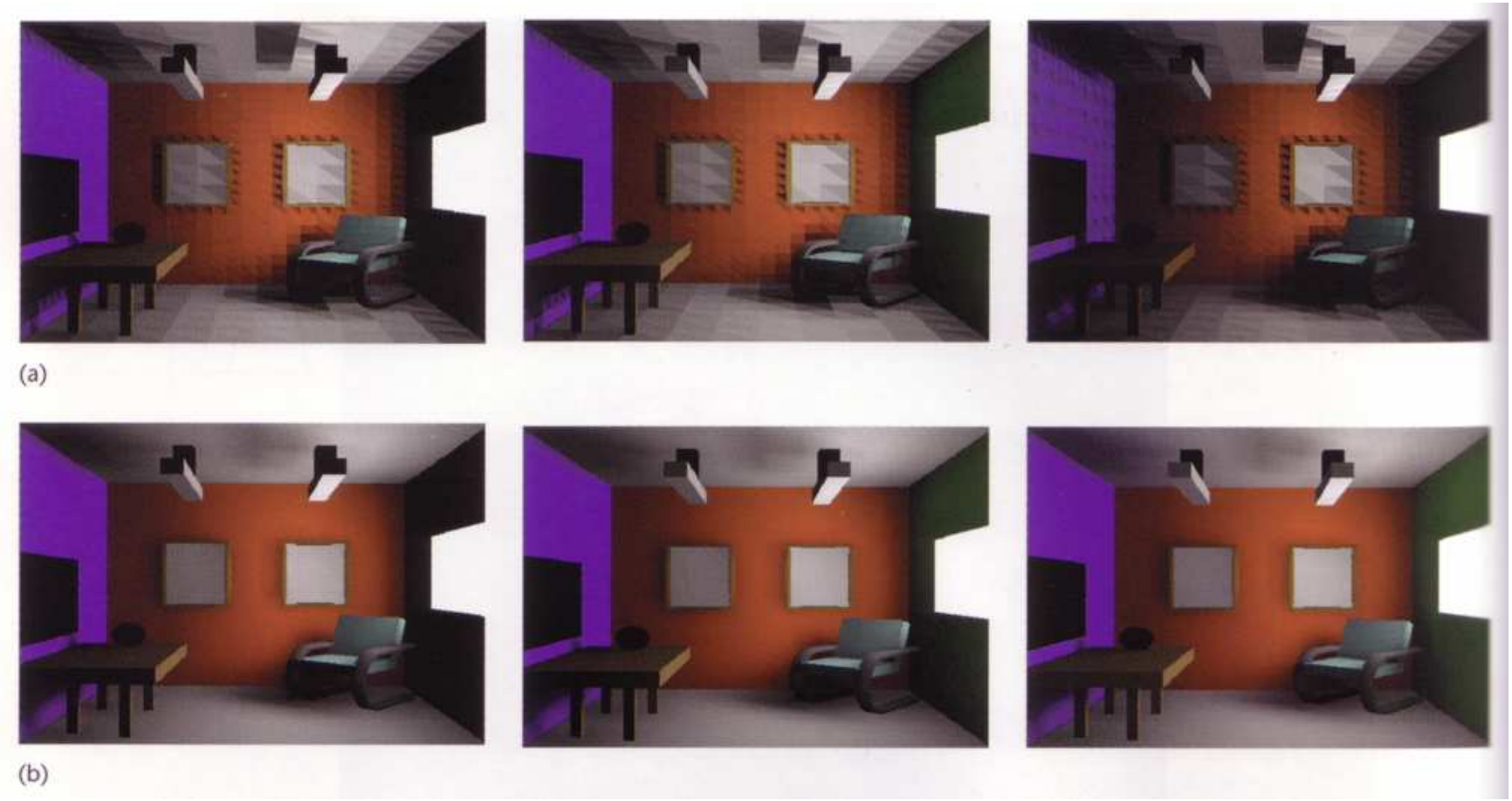
PROGRESSIVE SOLUTION

The above images show increasing levels of global diffuse illumination. From left to right: 0 bounces, 1 bounce, 3 bounces.

Progressive Radiosity

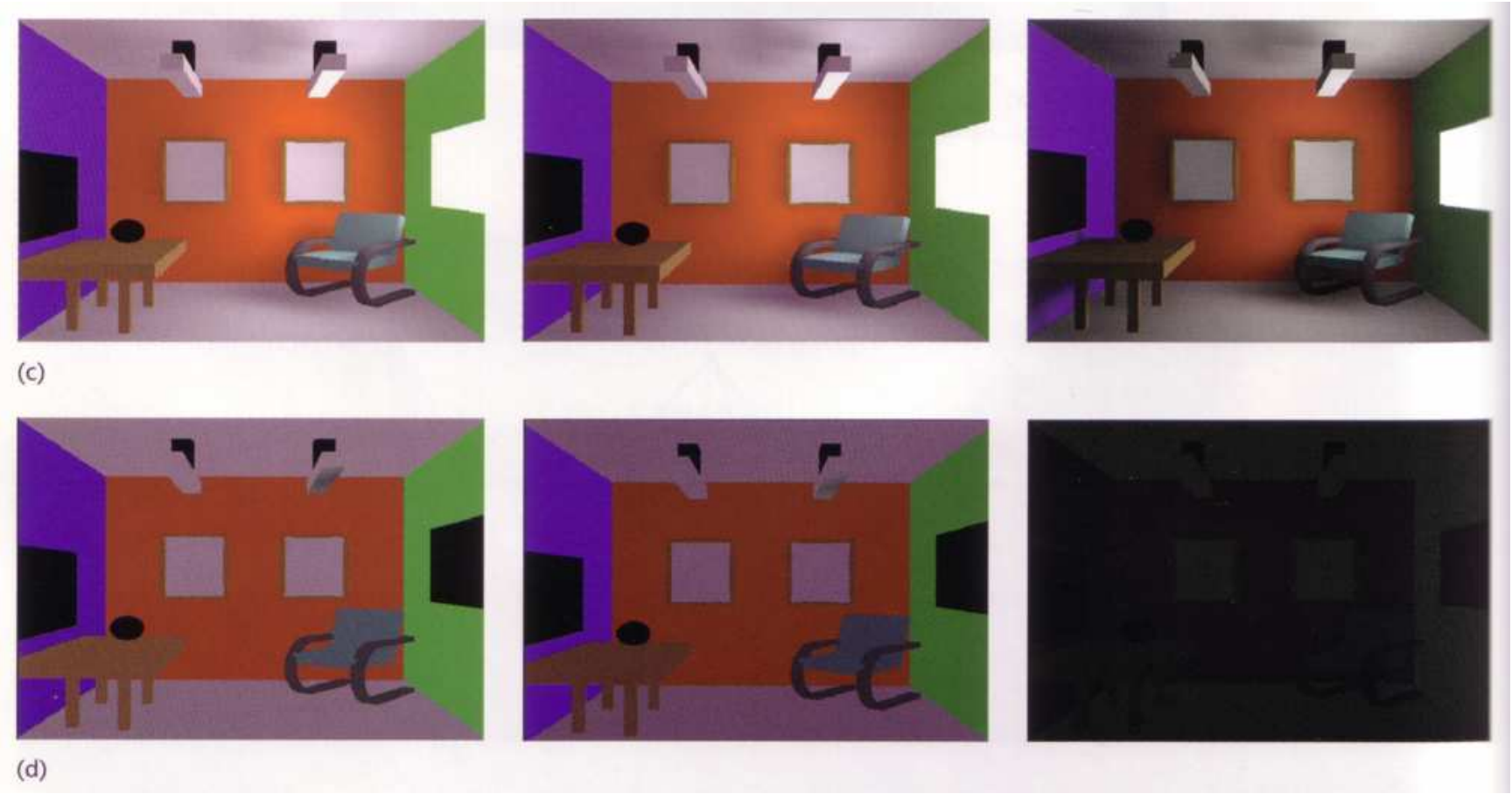


Progressive Radiosity



A radiosity image after 20, 250, and 5000 iterations of the progressive refinement method. From top to bottom for each column: (a) The radiosity solution as output from the iteration process. Each patch is allocated a constant radiosity. (b) The previous solution after it has been subjected to the interpolation process. (c) The same solution with the addition of the ambient term. (d) The difference between the previous two images. This gives a visual indication of the energy that had to be added to account for the unshot radiosity.

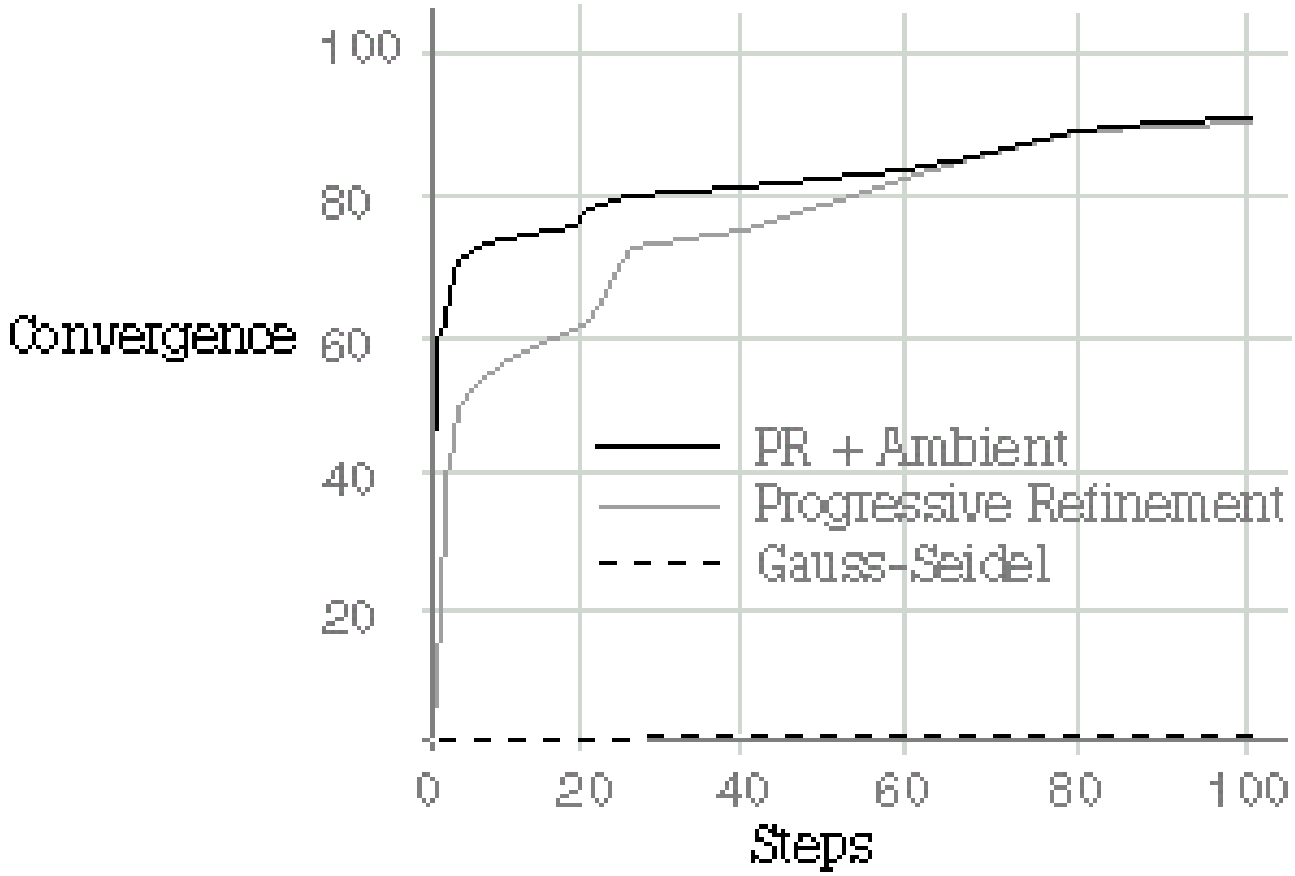
Effect of Ambient Light for Viewing



A radiosity image after 20, 250, and 5000 iterations of the progressive refinement method. From top to bottom for each column: (a) The radiosity solution as output from the iteration process. Each patch is allocated a constant radiosity. (b) The previous solution after it has been subjected to the interpolation process. (c) The same solution with the addition of the ambient term. (d) The difference between the previous two images. This gives a visual indication of the energy that had to be added to account for the unshot radiosity.

Progressive Radiosity

- Adaptive refinement



Yeap

PR: Some Special Cases

- Image after we have iterated through all light sources?

  - Shadows, but *no interreflections*

- Can incrementally display image while iterating
  - Add ambient light at each stage for visibility
  - Ambient shading if progressively refined
- Incremental form factor computation

Related: Stochastic Jacobi Radiosity

Algorithm 1 Incremental stochastic Jacobi iterative method.

1. Initialize total power $P_i \leftarrow P_{ei}$, unshot power $\Delta P_i \leftarrow P_{ei}$, and received power $\delta P_i \leftarrow 0$ for all patches i and compute total unshot power $\Delta P_T = \sum_i \Delta P_i$.
2. Until $\|\Delta P_i\| \leq \varepsilon$ or number of steps exceeds maximum, do
   (a) Choose number of samples N.
   (b) Generate a random number $\xi \in (0, 1)$.
   (c) Initialize $N_{prev} \leftarrow 0$; $q \leftarrow 0$.
   (d) Iterate over all patches i, for each i, do
      i. $q_i \leftarrow \Delta P_i / \Delta P_T$.
      ii. $q \leftarrow q + q_i$.
      iii. $N_i \leftarrow \lfloor Nq + \xi \rfloor - N_{prev}$.
      iv Do N_i times:
         A. Sample random point x on S_i.
         B. Sample cosine-distributed direction Θ at x.
         C. Determine patch j containing the nearest intersection point of the ray originating at x and with direction Θ, with the surfaces of the scene.
         D. Increment $\delta P_j \leftarrow \delta P_j + \frac{1}{N} \rho_j \Delta P_T$.
      v. $N_{prev} \leftarrow N_{prev} + N_i$.
   (e) Iterate over all patches i, increment total power $P_i \leftarrow P_i + \delta P_i$, replace unshot power $\Delta P_i \leftarrow \delta P_i$, and clear received power $\delta P_i \leftarrow 0$. Compute new total unshot power ΔP_T on the fly.
   (f) Display image using P_i.

Stochastic Jacobi Radiosity

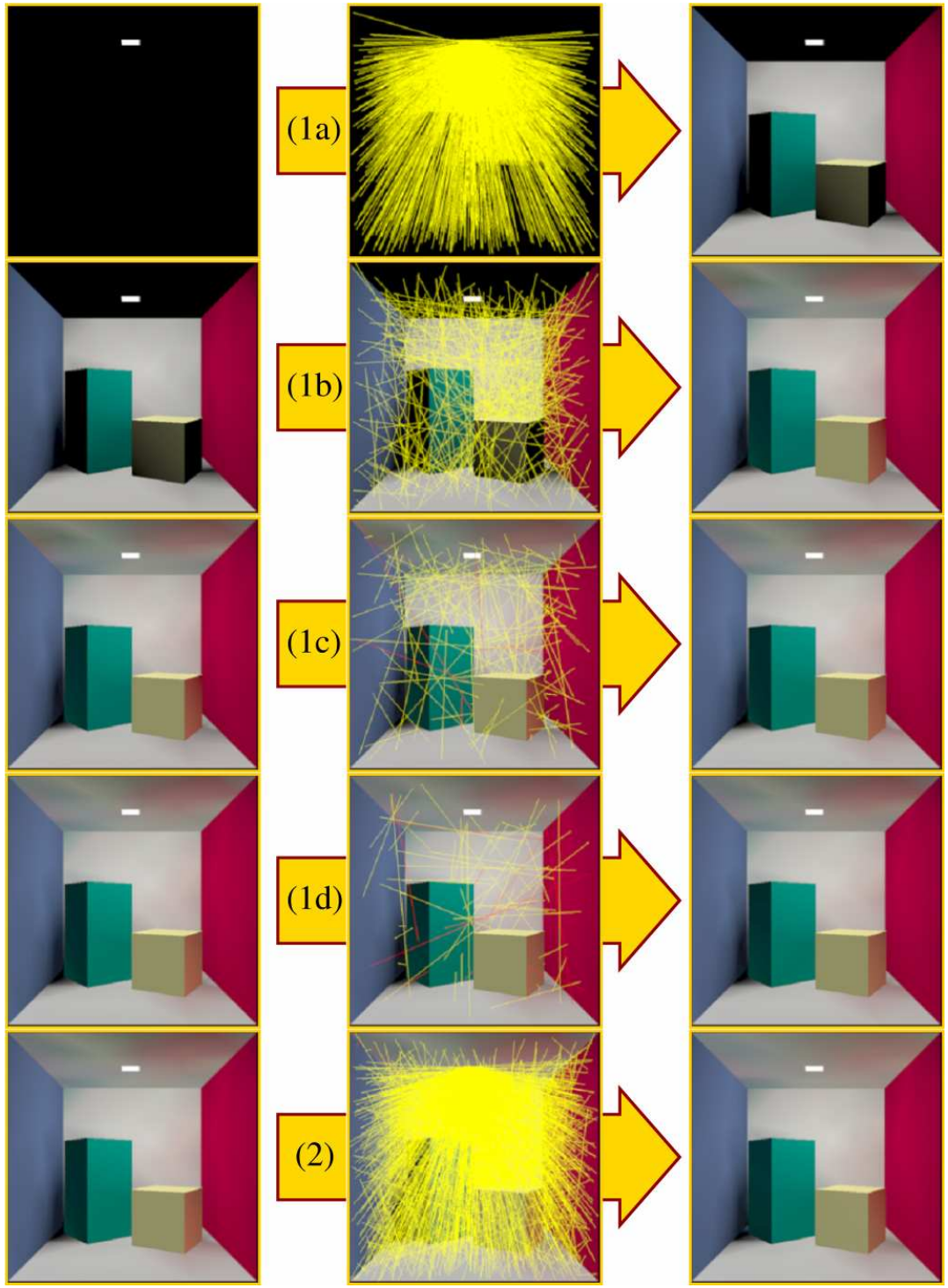


Fig 6.6, [Dutré et al. 2003]

1 iteration

4 iterations

16 iterations

64 iterations

252 iterations

Fig 6.7, [Dutré et al. 2003]

Overview

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

Surface Meshing

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

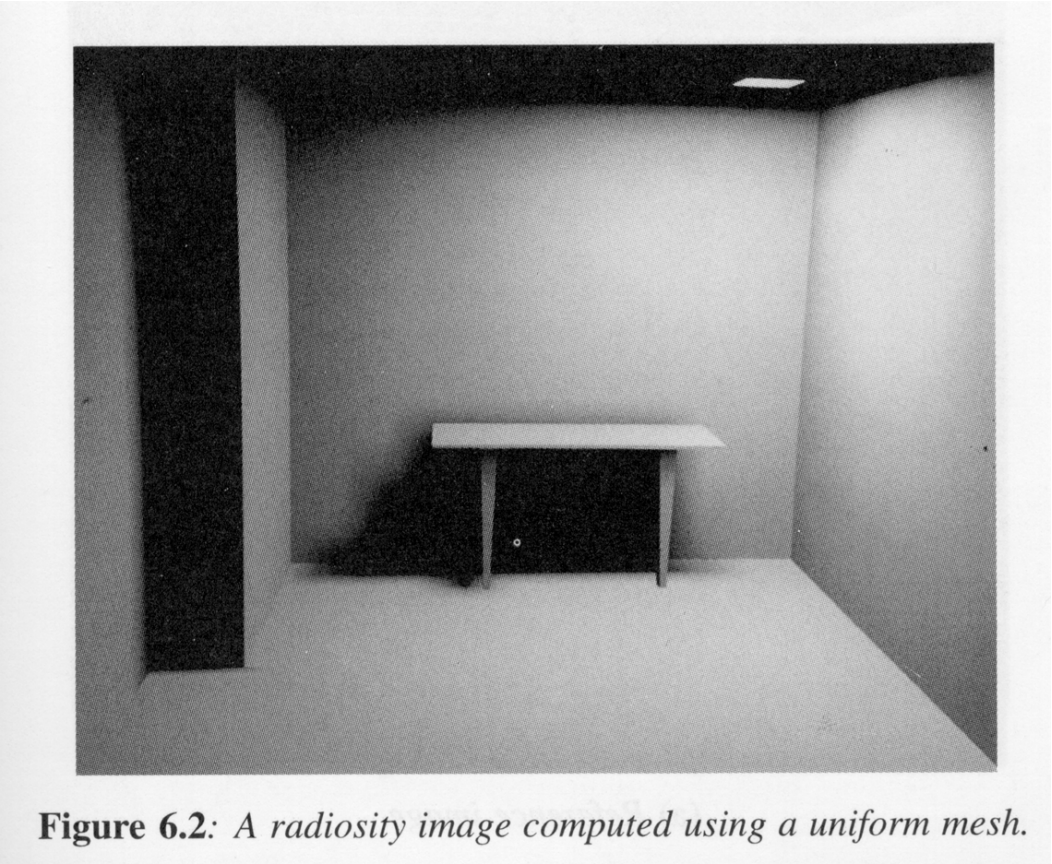
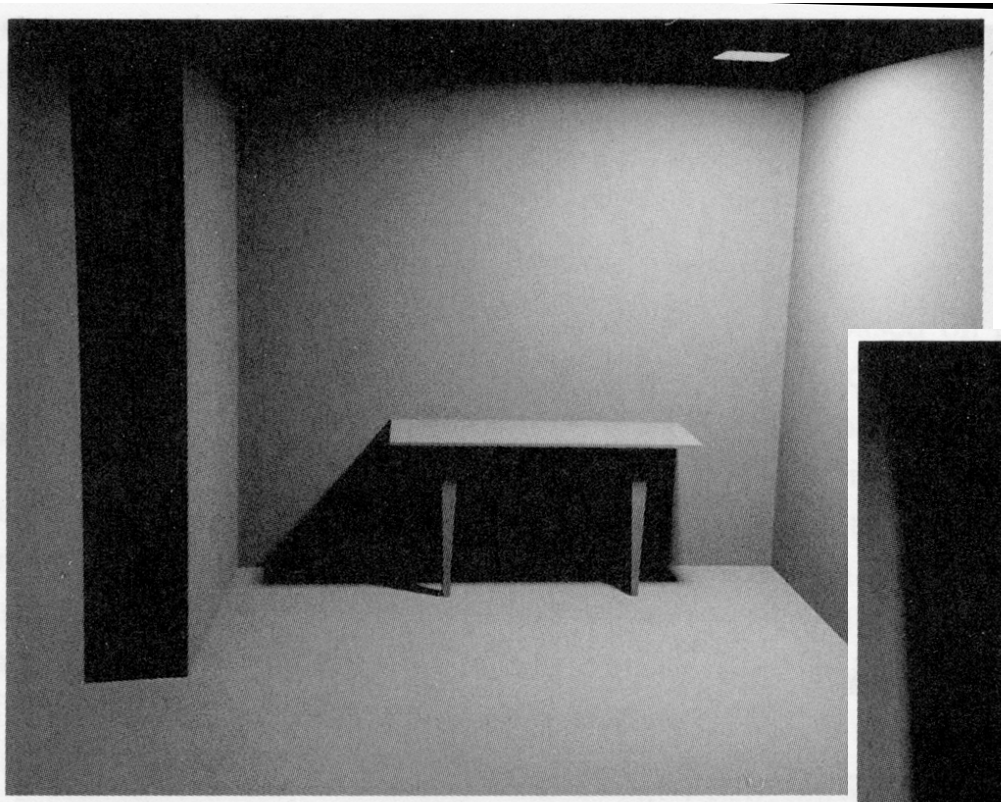


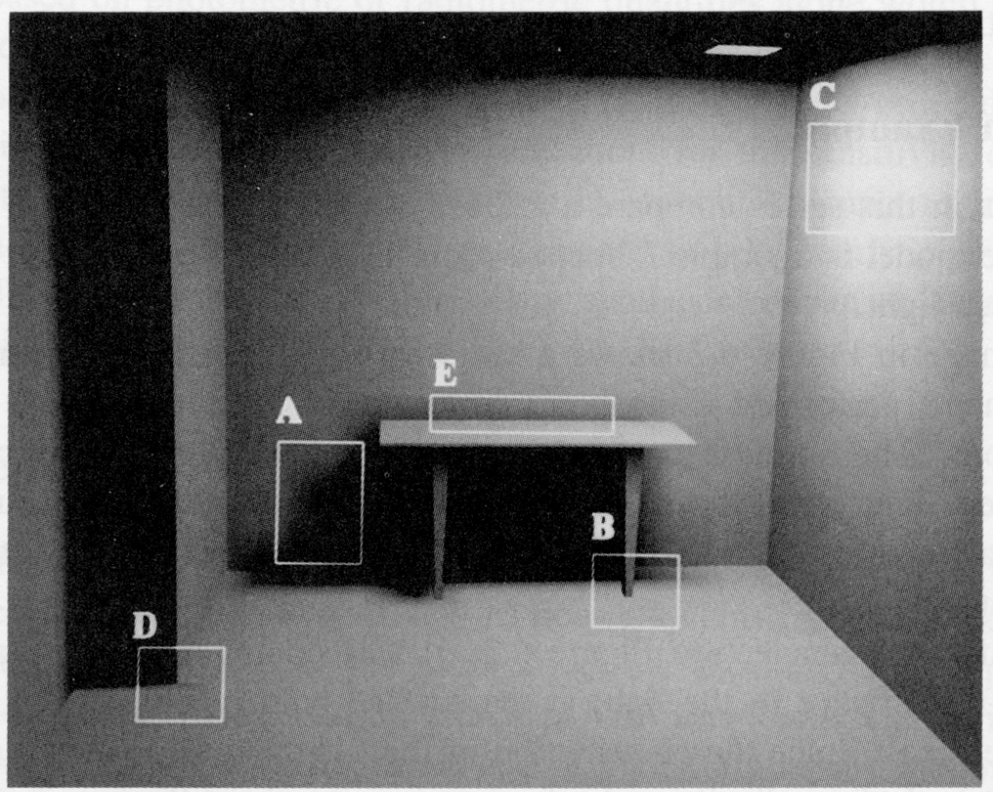
Figure 6.2: *A radiosity image computed using a uniform mesh.*

Cohen & Wallace

Artifacts of Bad Surface Meshing



(a) *Reference image.*



(b) *Artifacts introduced by the approximation.*

Cohen & Wallace

Error Image

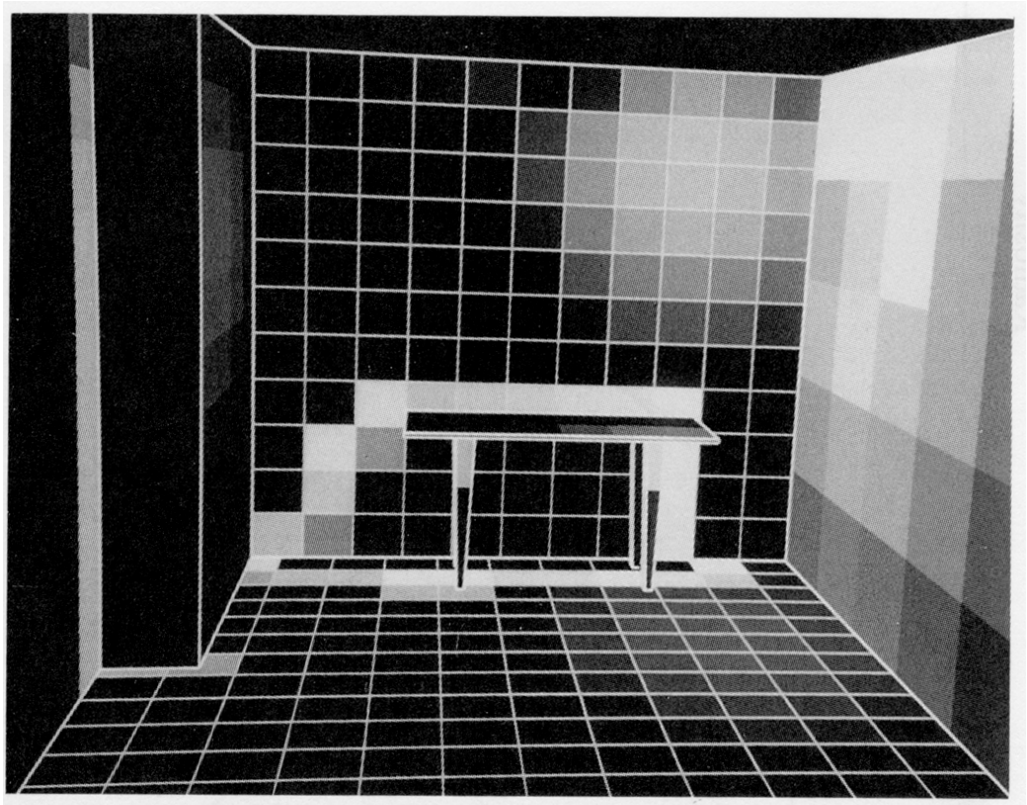
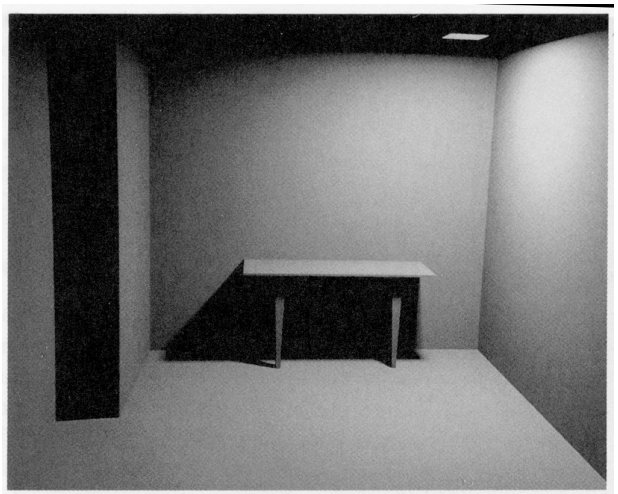
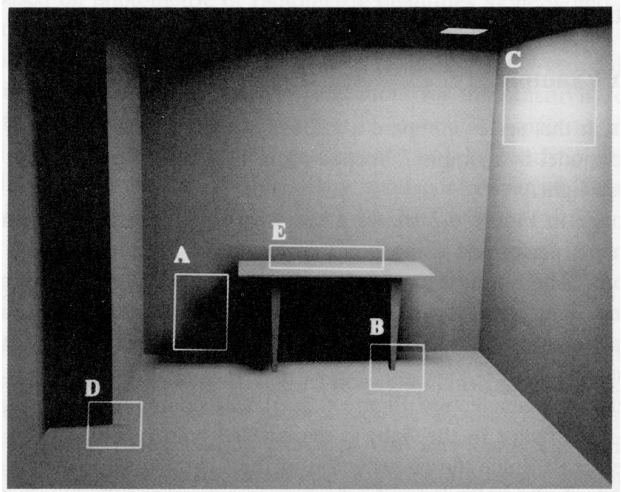


Figure 6.4: *Error image.*



(a) *Reference image.*



(b) *Artifacts introduced by the approximation.*

Cohen & Wallace

Error Image

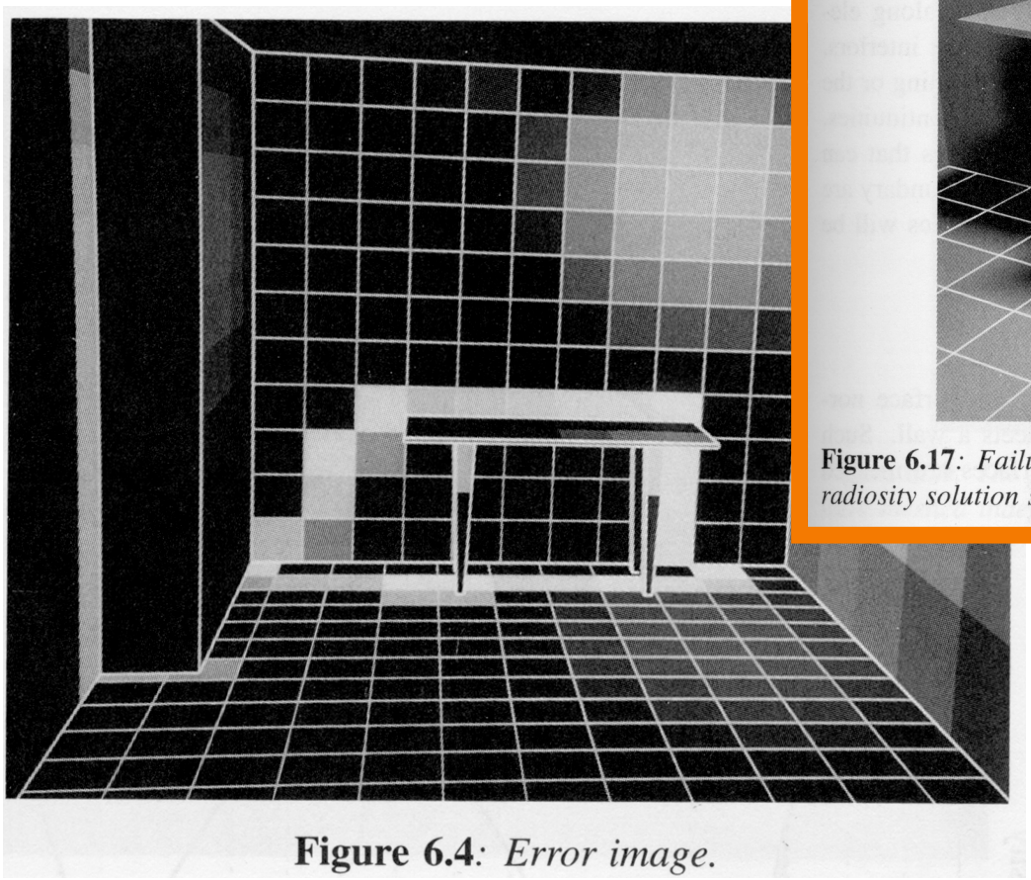


Figure 6.4: *Error image.*

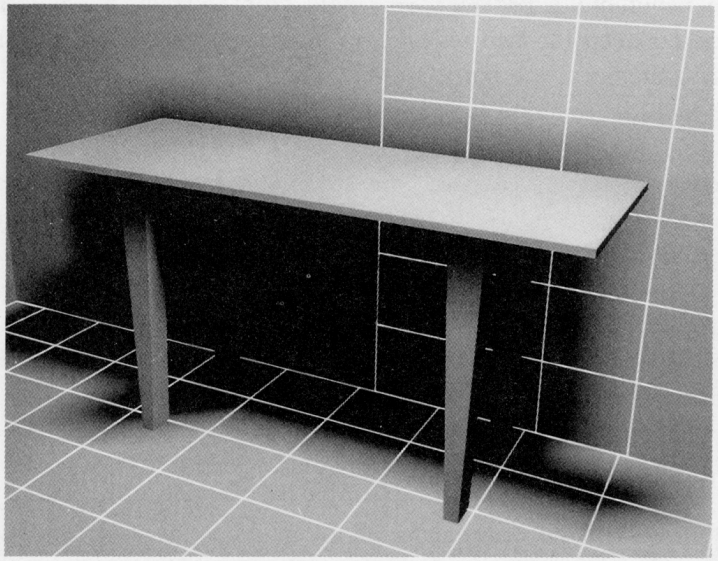
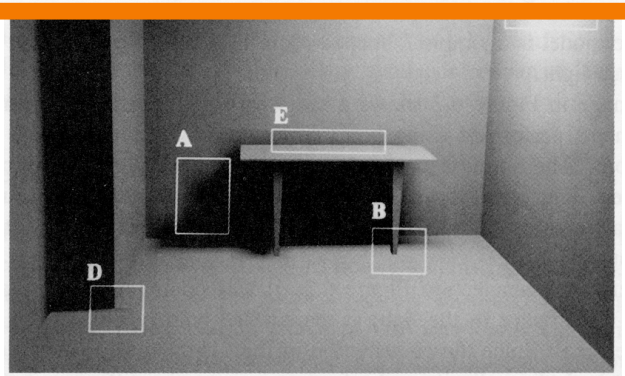


Figure 6.17: *Failure to resolve a discontinuity in value. This is a closeup of the radiosity solution shown in Figure 6.2.*

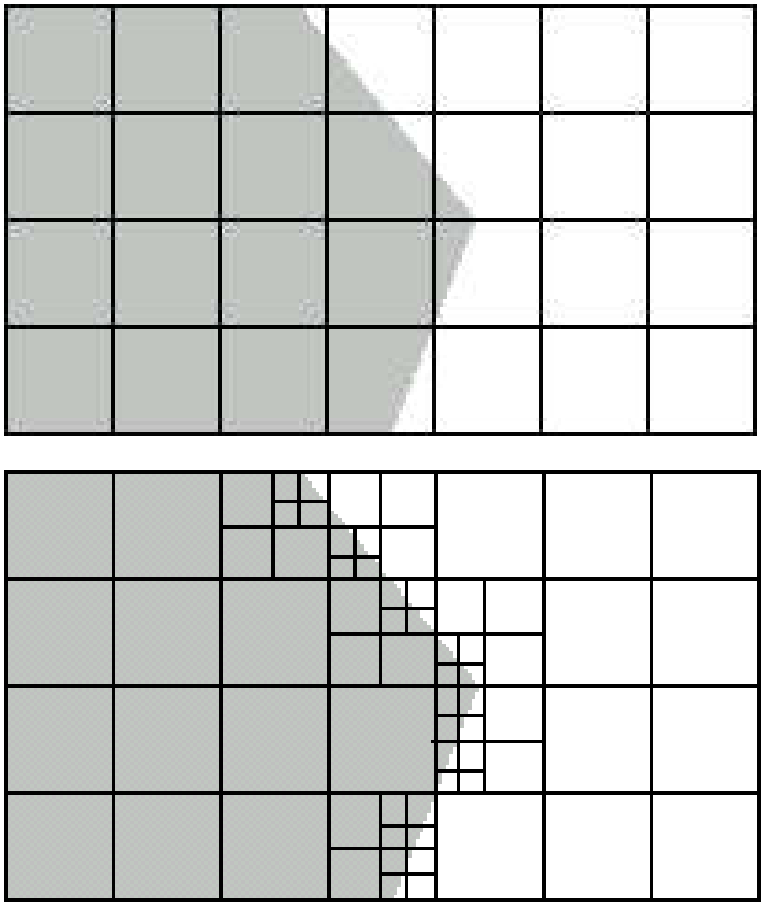


(b) *Artifacts introduced by the approximation.*

Cohen & Wallace

Adaptive Meshing

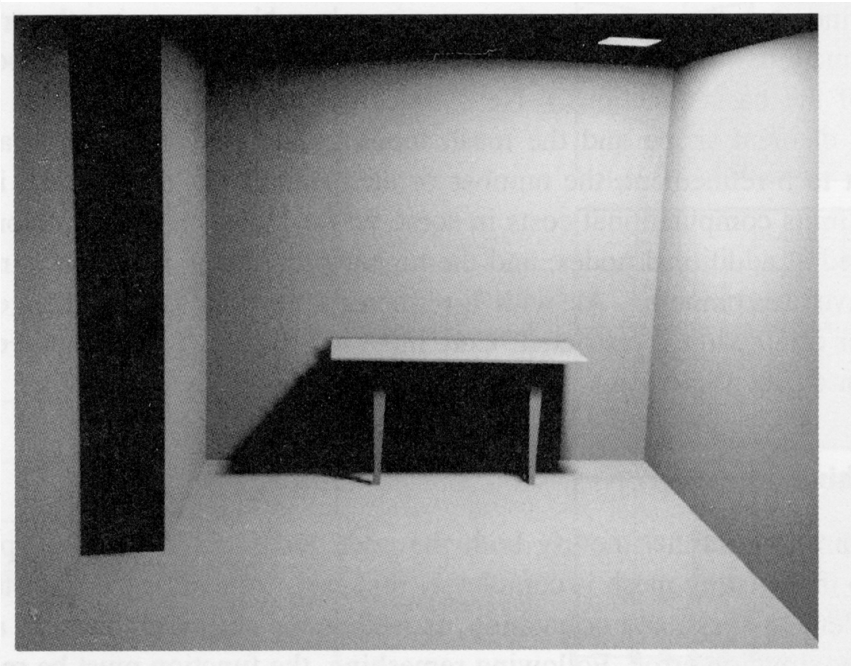
- Refine mesh in areas of high residual



Yeap

Adaptive Meshing

Uniform mesh



Adaptive mesh

Cohen & Wallace

Error Comparison

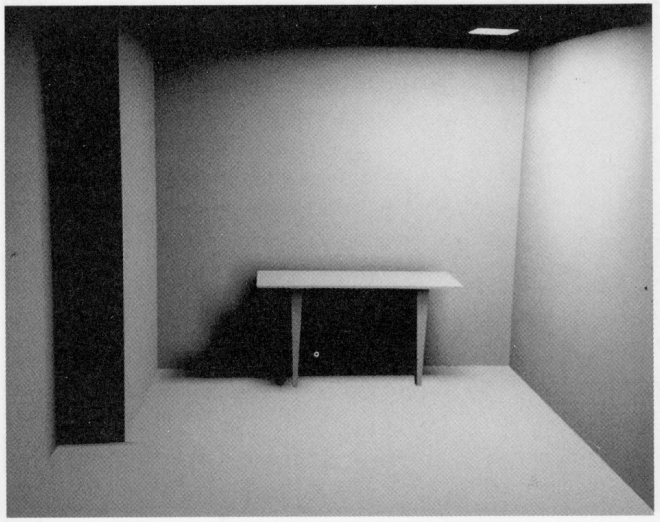


Figure 6.2: *A radiosity image computed using a uniform mesh.*

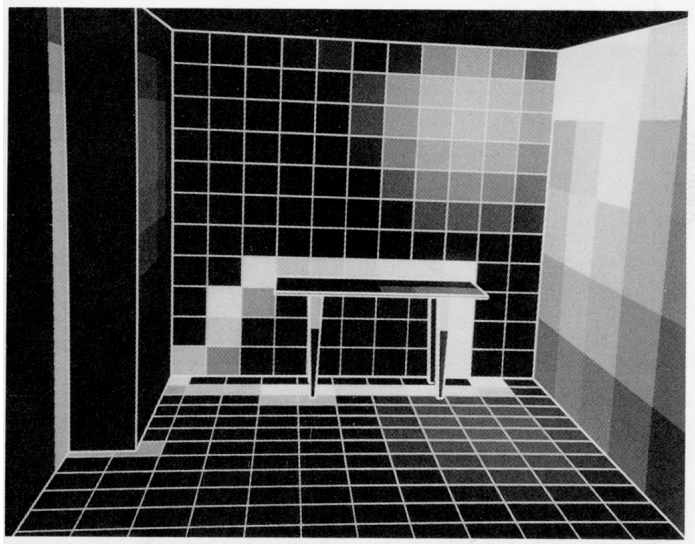


Figure 6.4: *Error image.*

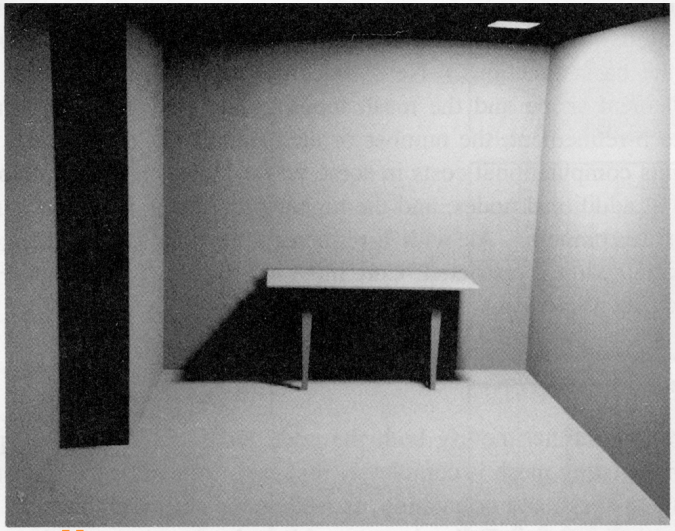


Figure 6.23: *Adaptive subdivision. Compare to Figure 6.2.*

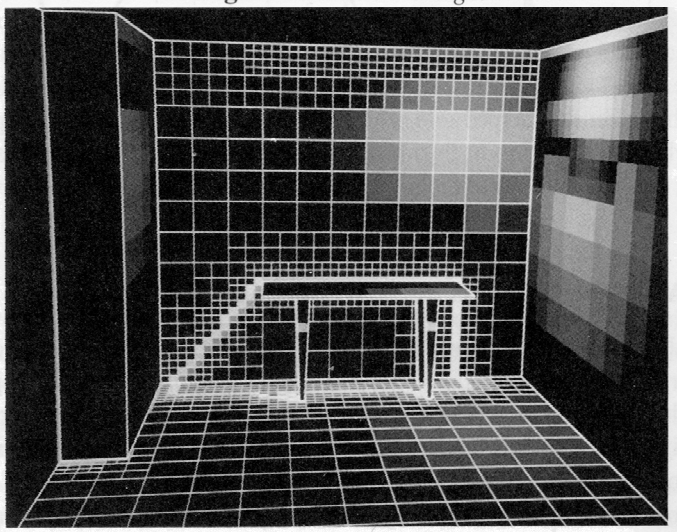
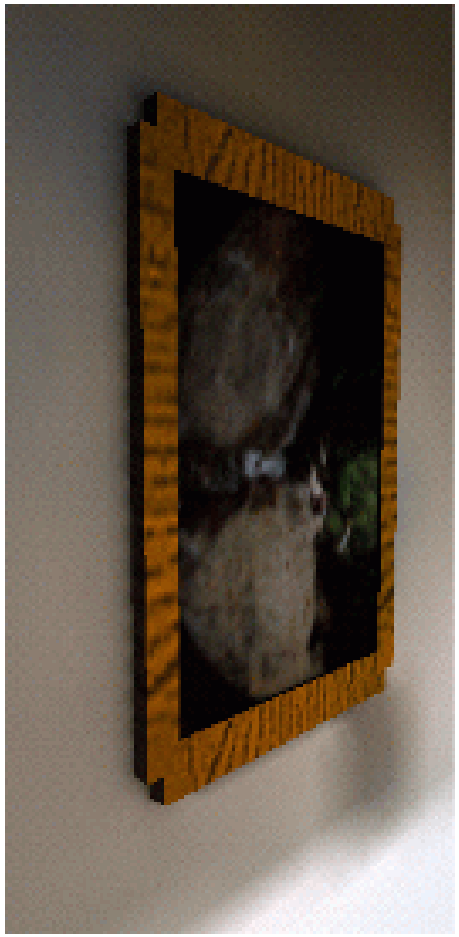
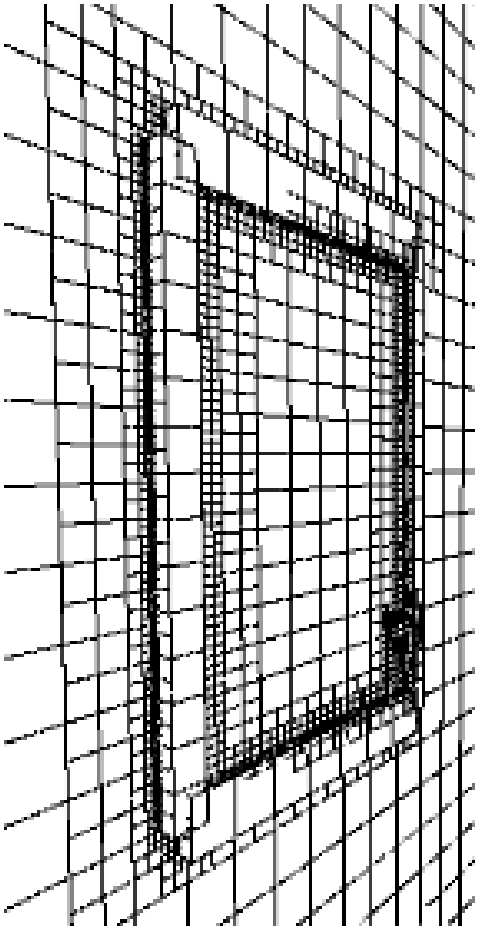


Figure 6.24: *Error image for adaptive subdivision. Compare to Figures 6.4 and*

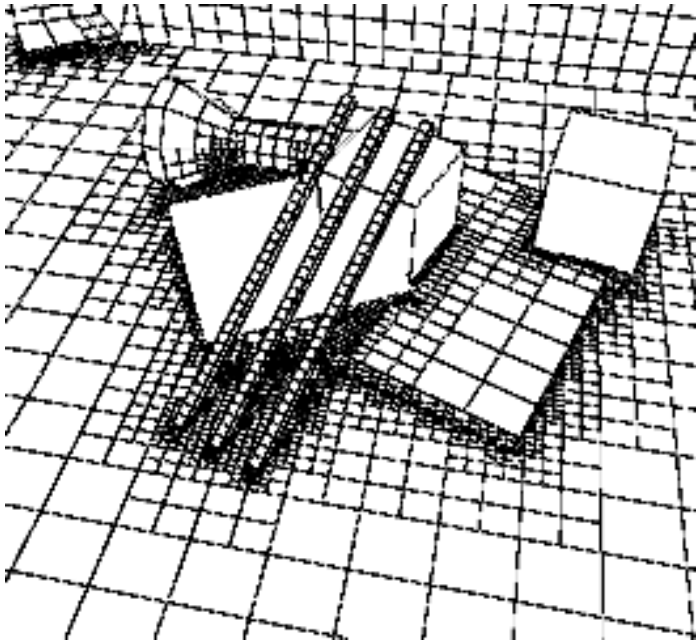
Cohen & Wallace

Adaptive Meshing





Adaptive Meshing



Adaptive Meshing

Baum et al.

Adaptive Meshing

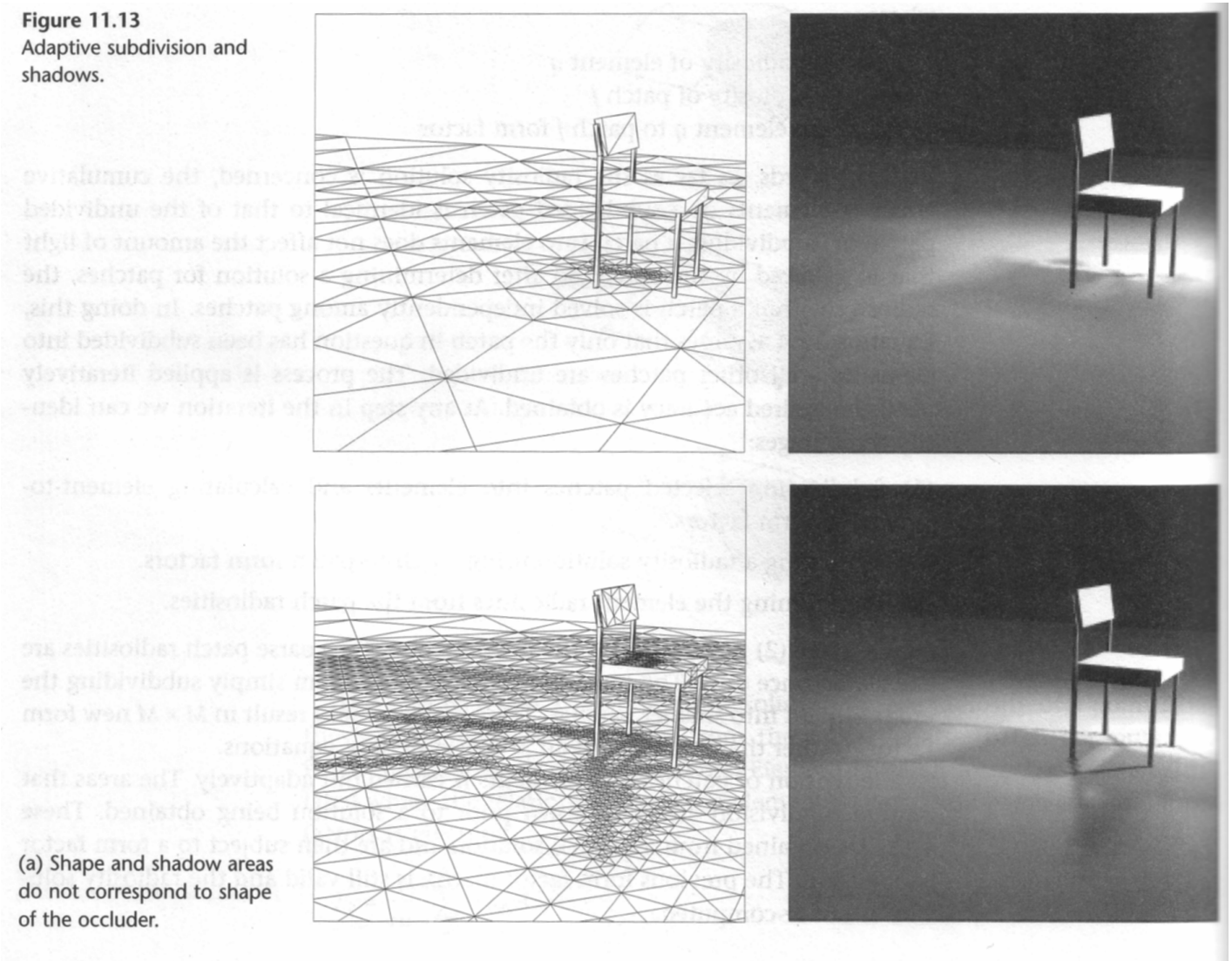


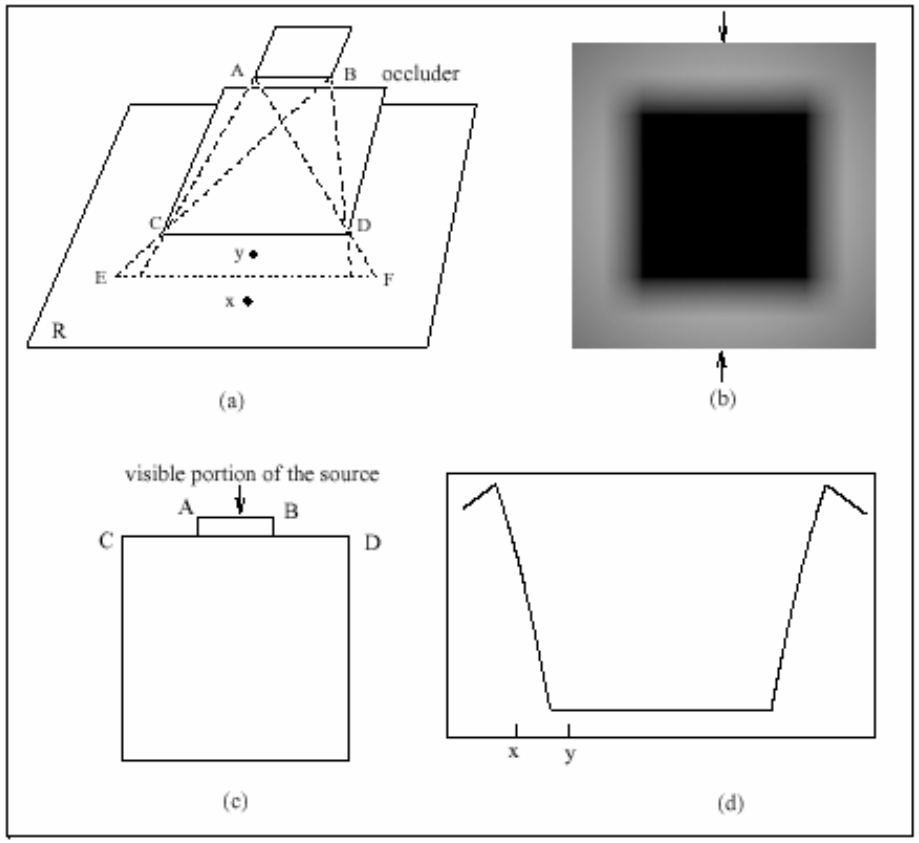
Figure 11.13
Adaptive subdivision and shadows.

(a) Shape and shadow areas do not correspond to shape of the occluder.

Watt, 3D CG, 3rd Ed, 2000

Discontinuity Meshing

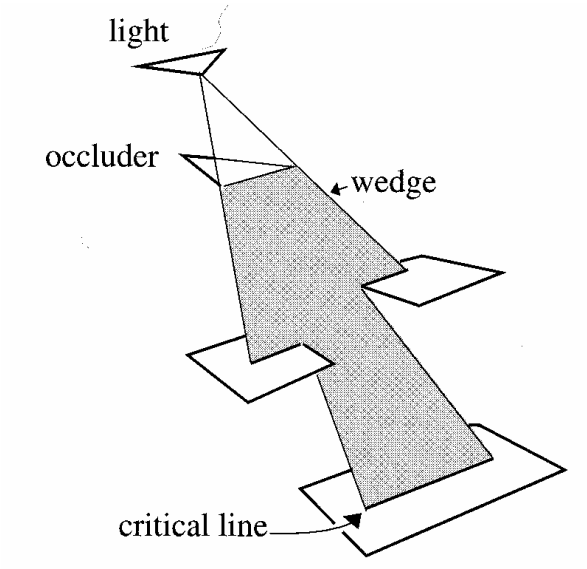
- Capture discontinuities in radiosity across a surface with explicit mesh boundaries



Lischinski et al.

Discontinuity Meshing

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

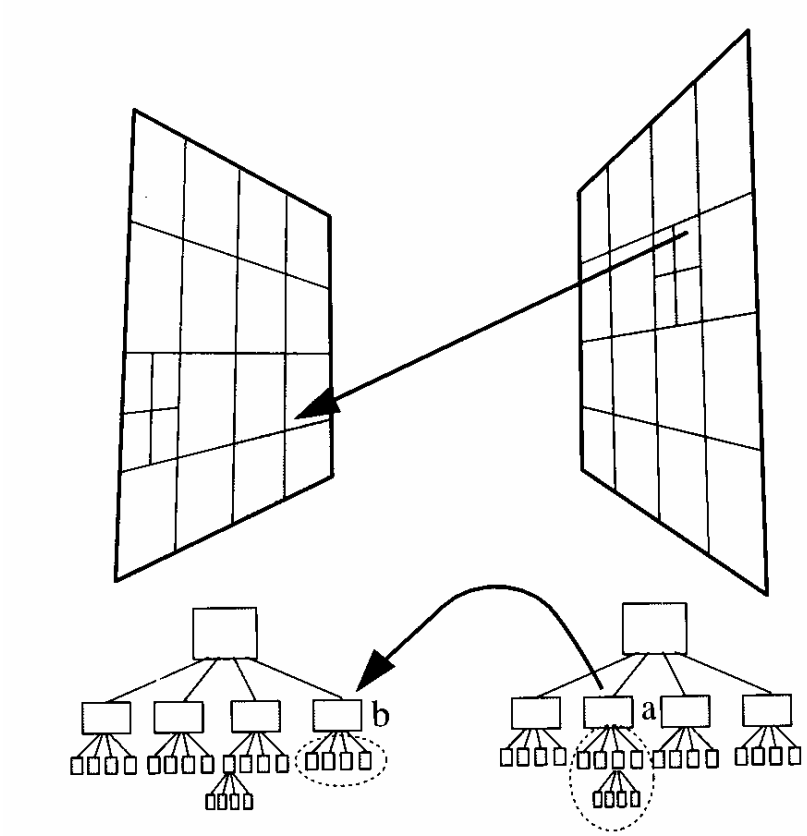


Discontinuity Mesh

Lischinski et al.

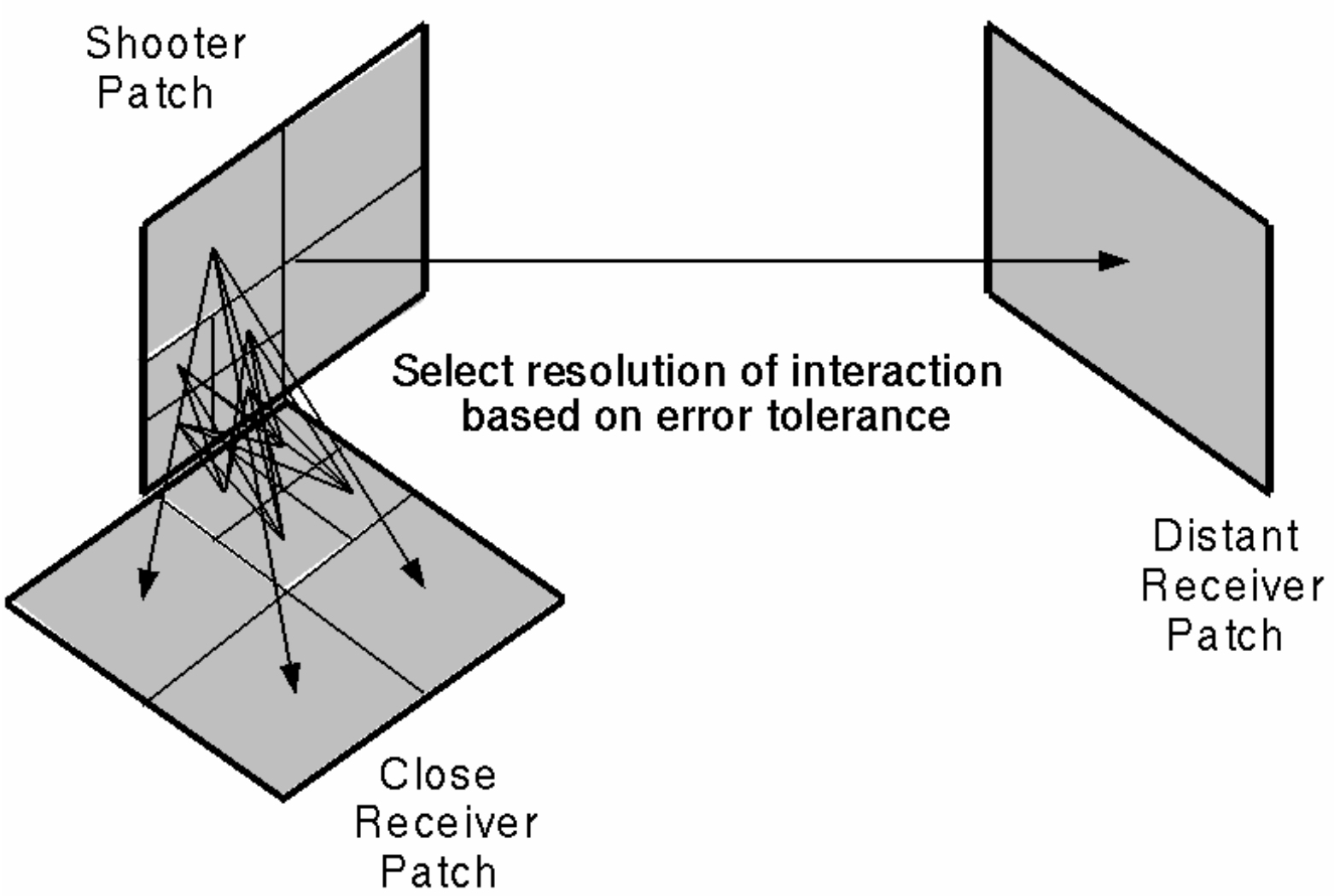
Hierarchical Radiosity

- Refine elements hierarchically:
  - Compute energy exchange at element granularity satisfying a user-specified error tolerance



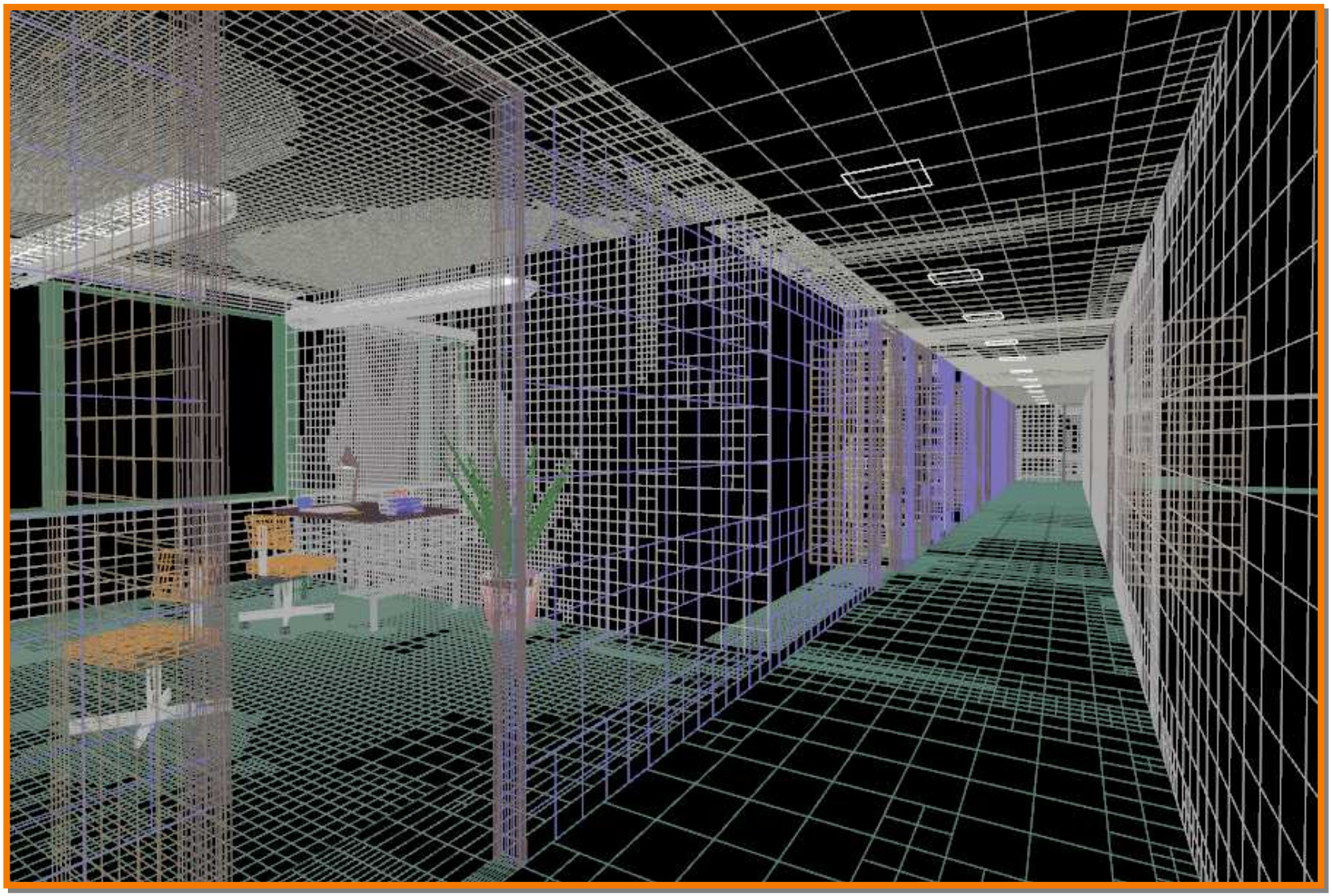
Cohen & Wallace

Hierarchical Radiosity



Hierarchical Radiosity

Hierarchical Radiosity



Hierarchical Radiosity

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

Example: Mirror Form Factors

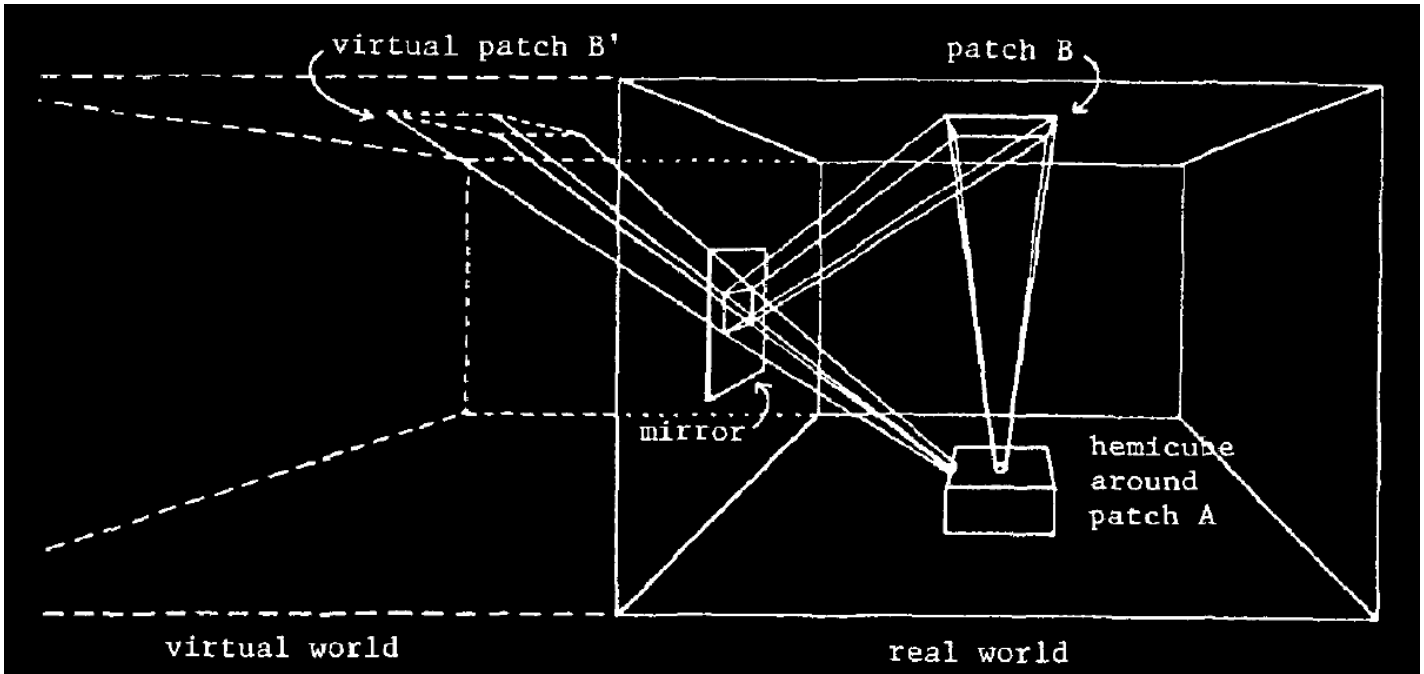


Figure 3. Calculation of extra form-factors to account for mirror reflection. Patch A receives light directly from patch B and indirectly through reflection by the mirror. The mirror is treated as a window into a virtual "mirror world." Projecting patch B′ onto the hemicube is then equivalent to following the actual path of reflection back to patch B.

Wallace, Cohen & Greenberg, 1987

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

Overview

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

Instant Radiosity

- Alexander Keller, SIGGRAPH 97
- “Particle-based” sampling method
  - Avoids discretization of underlying integral equation
  - Similar to photon mapping (density estimation) idea
- **Idea**: Hardware accelerated rendering of “photons” as point lights w/ shadows
- Limited number of photons due to sampling used
  - *Deterministic quasi-random walk* traces photons
    - » Jittered low-discrepancy sampling
- Fast and simple to implement

Basic Idea

- Render point-light sources into pixel (m,n) to evaluate integrals using graphics hardware

$$\langle L, \Psi \rangle := \int_S \int_\Omega L(y, \vec{\omega}) \Psi(y, \vec{\omega}) \cos\theta d\omega dy$$

$$\Psi_{mn}(y, \vec{\omega}) := \frac{\delta(\vec{\omega} - \vec{\omega}_{y_f})}{\cos\theta} \frac{1}{|P_{nm}|} \chi_{P_{mn}}(h(y, \vec{\omega}))$$

- How? Use M point lights $L(y) \approx \sum_{i=0}^{M-1} L_i \delta(y - P_i),$
- → Integrals become trivial to evaluate

$$\begin{aligned} \overline{L}_{mn} := \langle L, \Psi_{mn} \rangle &= \langle L_e, \Psi_{mn} \rangle + \langle T_{f_r} L, \Psi_{mn} \rangle \\ &= \langle L_e, \Psi_{mn} \rangle + T_{mn} L \end{aligned}$$

Sampling: Choose the particles wisely

3.1 Quasi-Monte Carlo Integration

Similar to [Kel96b], we use the method of quasi-Monte Carlo integration for the evaluation of the integrals. For the approximation

$$\int_{[0,1)^s} f(x)dx \approx \frac{1}{N}\sum_{i=0}^{N-1} f(x_i)$$

we generate the N sample points $x_0, \ldots, x_{N-1} \in [0,1)^s$ using the Halton sequence. Based on the radical inverse function

$$\Phi_b(i) := \sum_{j=0}^{\infty} a_j(i)\; b^{-j-1} \in [0,1) \Leftrightarrow i = \sum_{j=0}^{\infty} a_j(i)\; b^j,$$

the s-dimensional Halton low discrepancy sequence (see comparison in figure 1) is

$$x_i = \left(\Phi_{b_1}(i), \ldots, \Phi_{b_s}(i)\right), i \in \mathbb{N}_0,$$

where b_j is the j-th prime number. Note that each segment $P_{N'}$ of

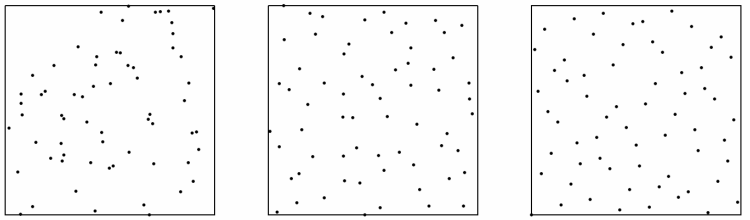


Figure 1: Two-dimensional uniform sampling patterns: a) random, b) jittered, and c) Halton for $N = 64$ samples.

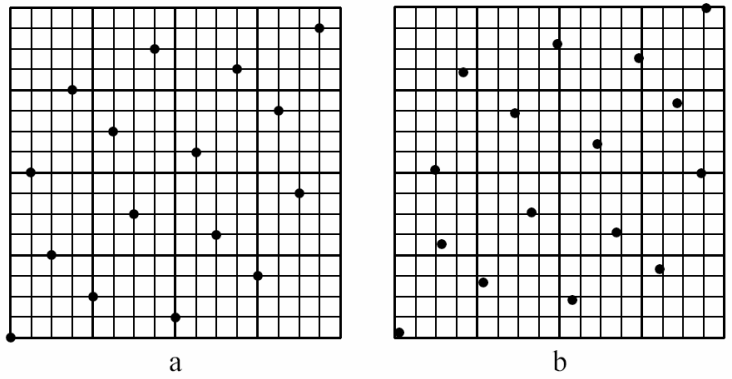


Figure 5: Grid structure of the a) Hammersley and b) jittered Hammersley sampling patterns for $N = 16$.

Sampling

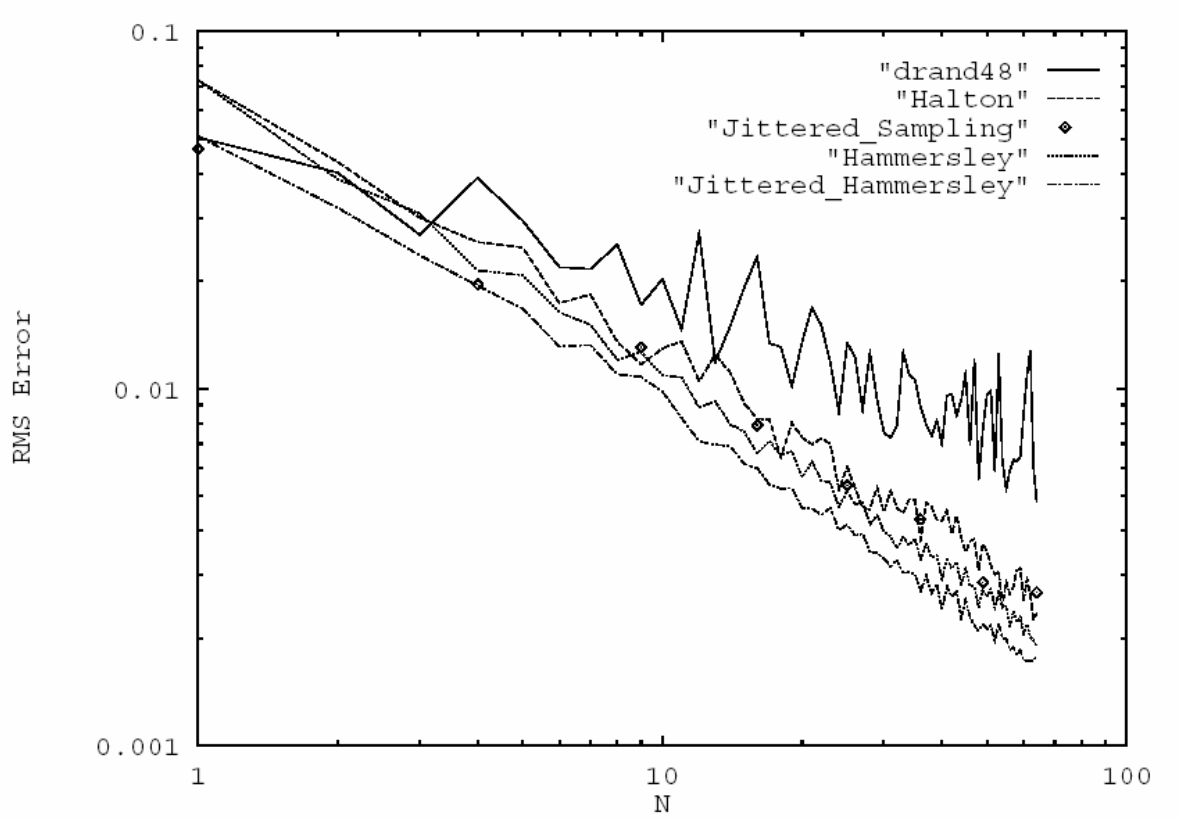


Figure 6: Convergence of different sampling patterns.

Quasi-Random Walk

$$L = (I - T_{f_r})^{-1} L_e = \sum_{j=0}^{\infty} T_{f_r}^j L_e$$

converges and can be used to solve the integral equation. Inserted in (1), after some transformations, for the radiosity setting we get

$$T_{mn} L = \frac{1}{|P_{mn}|} \sum_{j=0}^{\infty} \int_{P_{mn}} \int_{\Omega^j} \int_{S_e} p_j(y_0, \vec{\omega}_0, \ldots, \vec{\omega}_j)$$

$$V(y_j, y') f_d(y') \frac{\cos\theta_j \cos\theta'}{|y_j - y'|^2} dy_0 d\omega_0 \cdots d\omega_j dP. \quad (3)$$

Here $S_e := \text{supp } L_e \subseteq S$ is the support of the light sources, and $y' = h(y_f, P - y_f) \in S$ is the first point hit when shooting a ray from the eye at y_f through the point $P \in P_{mn}$ into the scene. $V(y_j, y')$ checks the mutual visibility of the points y_j and y', yielding 1 in case of visibility and 0 else. The radiance density

$$p_j(y_0, \vec{\omega}_0, \ldots, \vec{\omega}_j) := L_e(y_0) \prod_{l=1}^{j} (\cos\theta_{l-1} f_d(y_l))$$

Quasi-Random Walk

- Assumption: mean reflectivity representative of local reflectivity in scene

$$\overline{\rho} := \frac{\sum_{k=1}^{K} \rho_{d,k} |A_k|}{\sum_{k=1}^{K} |A_k|} \approx \|T_{f_d}\|,$$

- Use mean reflectivity to estimate number of particles reflected at each generation
- Total number of particles is bounded by

$$M < \sum_{j=0}^{\infty} \overline{\rho}^j N = \frac{1}{1-\overline{\rho}} N =: \bar{l} N$$

Pseudocode

```
void InstantRadiosity(int N, double ρ̄)
{
    double w, Start; int End, Reflections = 0;
    Color L; Point y; Vector ω⃗;

    Start = End = N;

    while(End > 0)
    {
        Start *= ρ̄;

        for(int i = (int) Start; i < End; i++)
        {
            // Select starting point on light source
            y = y_0(Φ_2(i), Φ_3(i));
            L = L_e(y) * supp L_e;
            w = N;

            // trace reflections
            for(int j = 0; j <= Reflections; j++)
            {
                glRenderShadowedScene(N/⌊w⌋ L, y);
                glAccum(GL_ACCUM, 1/N);
                // diffuse scattering
                ω⃗ = ω⃗_d(Φ_{b_{2j+2}}(i), Φ_{b_{2j+3}}(i));
                //trace ray from y into direction ω⃗
                y = h(y, ω⃗);
                // Attenuate and compensate
                L *= f_d(y);
                w *= ρ̄;
            }
        }

        Reflections++;
        End = (int) Start;
    }

    glAccum(GL_RETURN, 1.0);
}
```

Algorithm (N=10, M=20)

$\langle L_e, \Psi_{mn} \rangle$
$+T_{mn}L_e$

$+T_{mn}T_{f_d}L_e$

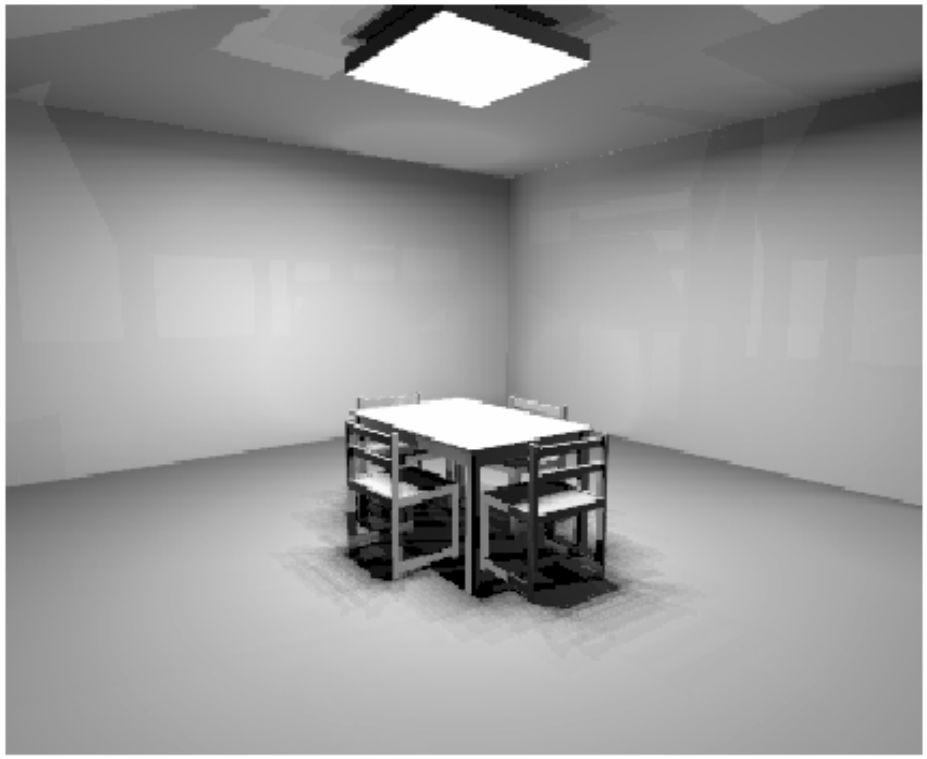
$+T_{mn}T_{f_d}^2L_e$

$+T_{mn}T_{f_d}^3L_e$

$+T_{mn}T_{f_d}^4L_e$

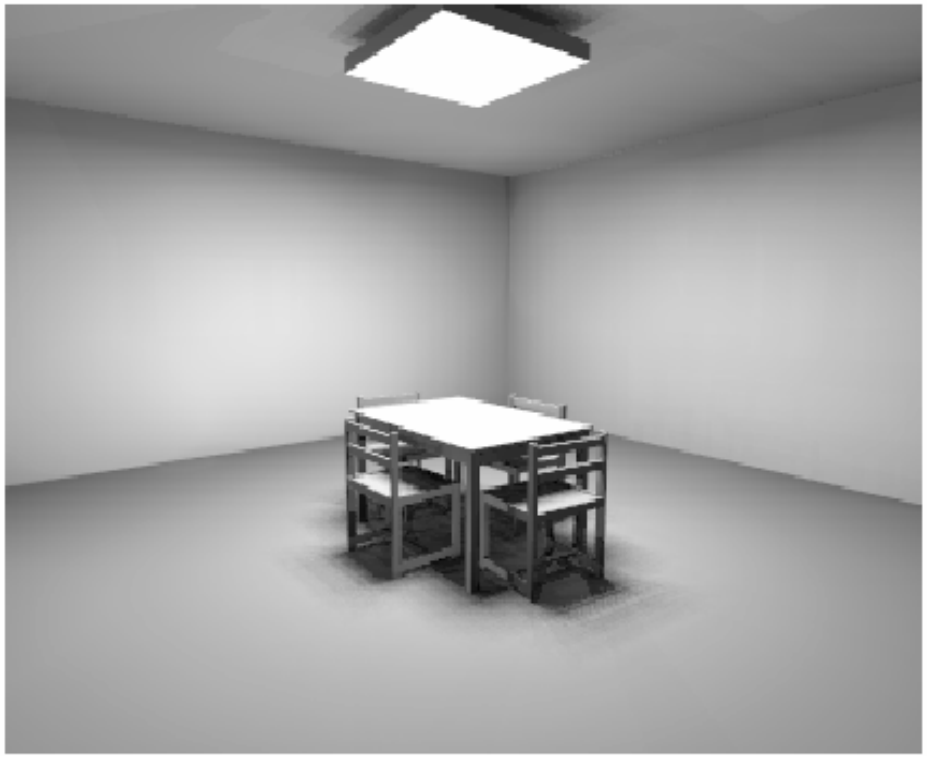
Results

$N = 10$
$M = 20$



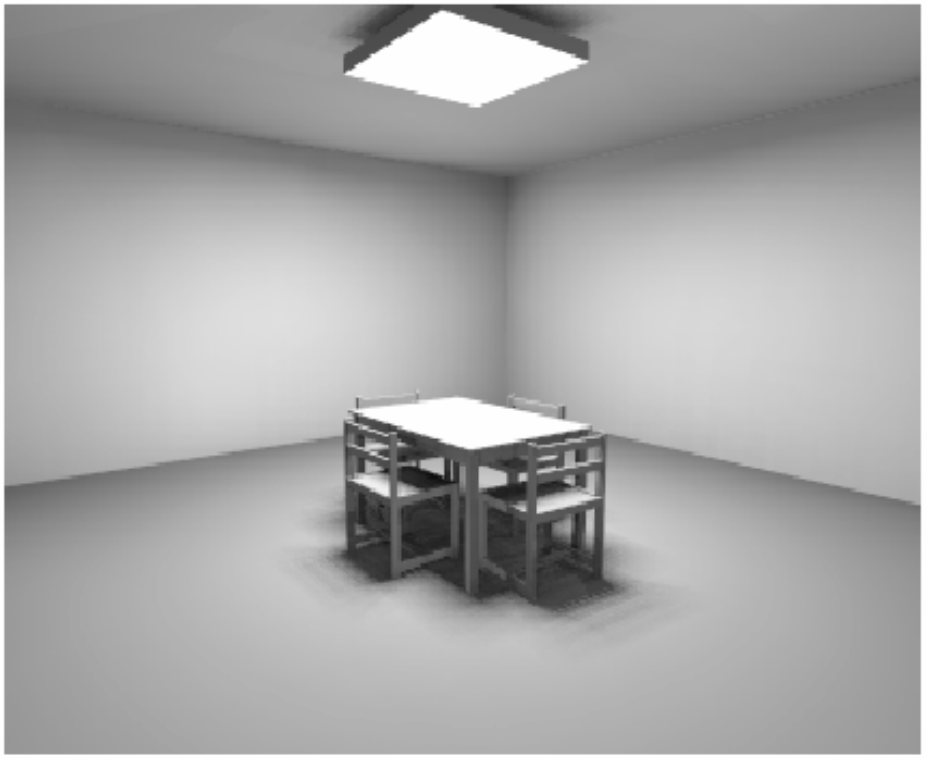
Results

$N = 32$
$M = 72$



Results

$N = 64$
$M = 147$



Results

Figure 8: Specular effects of the standlight on the floor by using the full BRDF f_r in T_{mn} for $N = 128$.

Results

Figure 9: Conference room image for $N = 128$.