15-462 Computer Graphics I

Lecture 19

# Global Illumination

Substructuring
Progressive Refinement
Bidirectional Reflectance Dist. Fcn.
Combining Radiosity and Ray Tracing

[Angel, Ch 13.5]

November 13, 2003

Doug James
Carnegie Mellon University

http://www.cs.cmu.edu/~djames/15-462/Fall03

# Classical Radiosity Method

- Divide surfaces into patches
- Model light transfer between patches as system of linear equations
- Important assumptions (so far):
  - Reflection and emission are diffuse
  - No participating media (no fog)
  - No transmission (only opaque surfaces)
  - Radiosity is constant across each patch
  - Solve for R, G, B separately

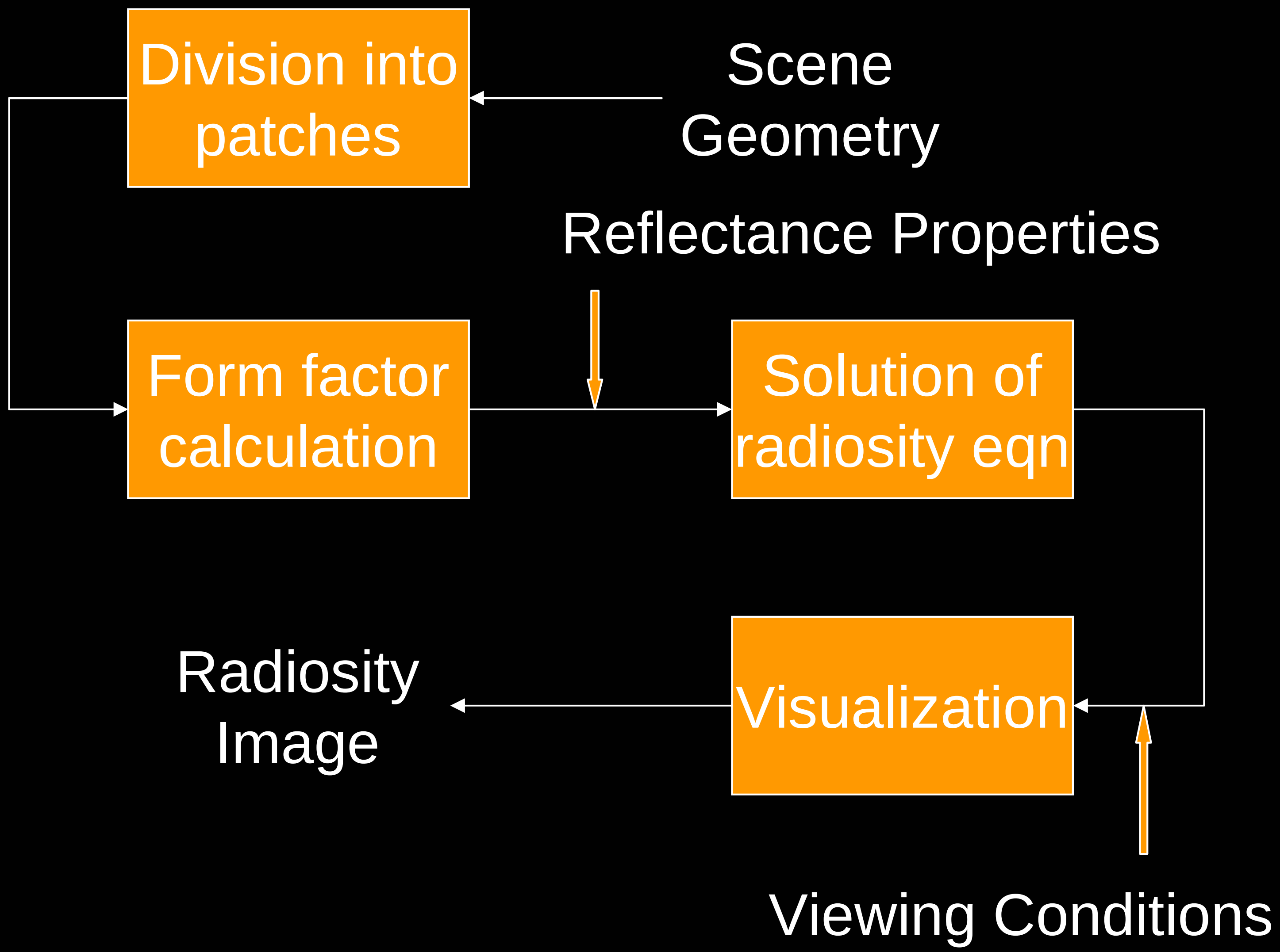
# Radiosity Equation

- For each patch i:

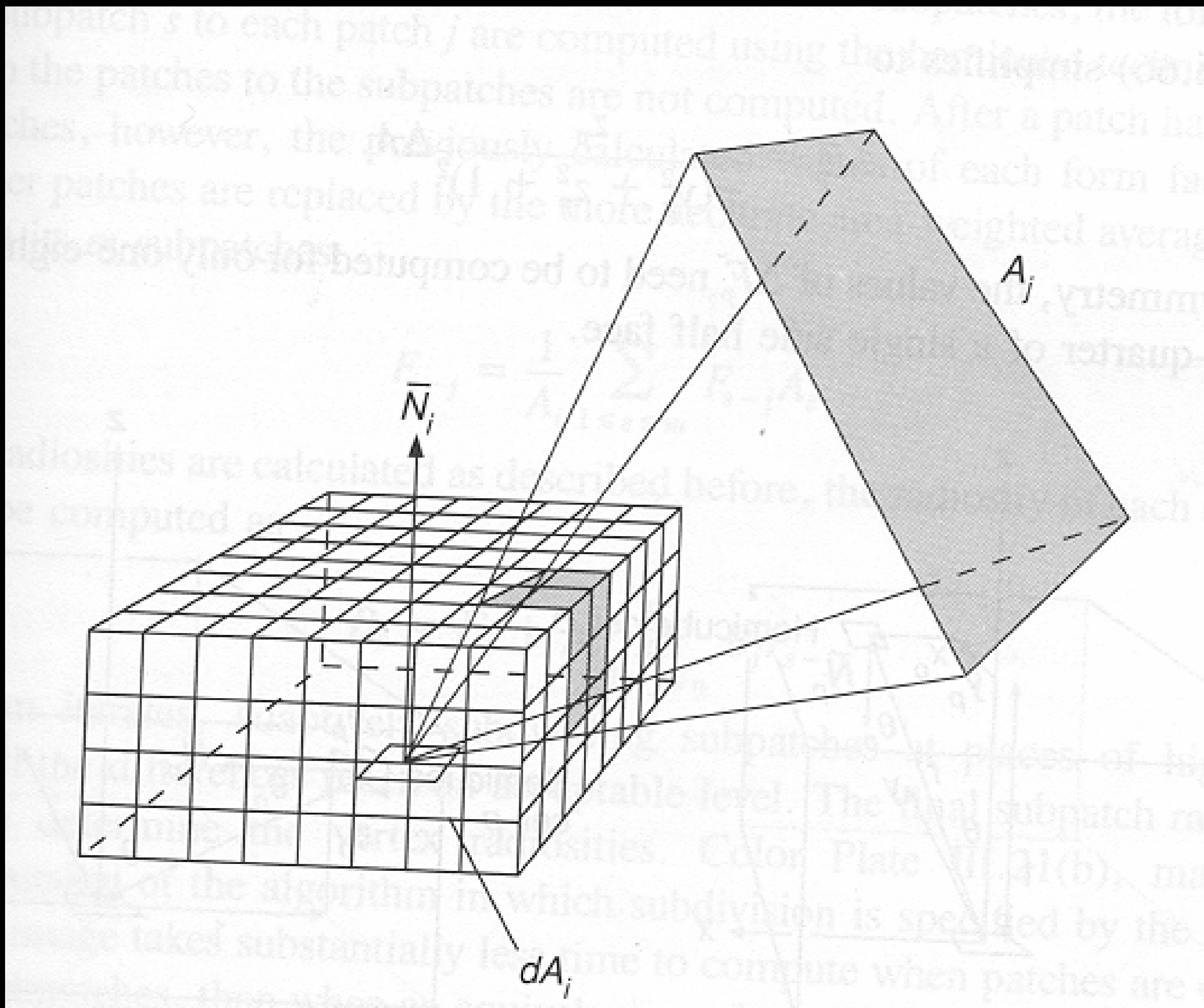
$$\begin{aligned} B_i &= E_i + \rho_i \sum_j (F_{ji} A_j / A_i) B_j \\ &= E_i + \rho_i \sum_j F_{ij} B_j \end{aligned}$$

- Variables
  - $B_i$ = radiosity (unknown)
  - $E_i$ = emittance of light sources (given)
  - $\rho_i$ = reflectance (given)
  - $F_{ij}$ = form factor from i to j (computed)
    fraction of light emitted from patch i arriving at patch j
  - $A_i$ = area of patch i (computed)

# Idealized Radiosity Computation



# Form Factors via Hemicubes




R. Ramamoorthi

# Outline

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

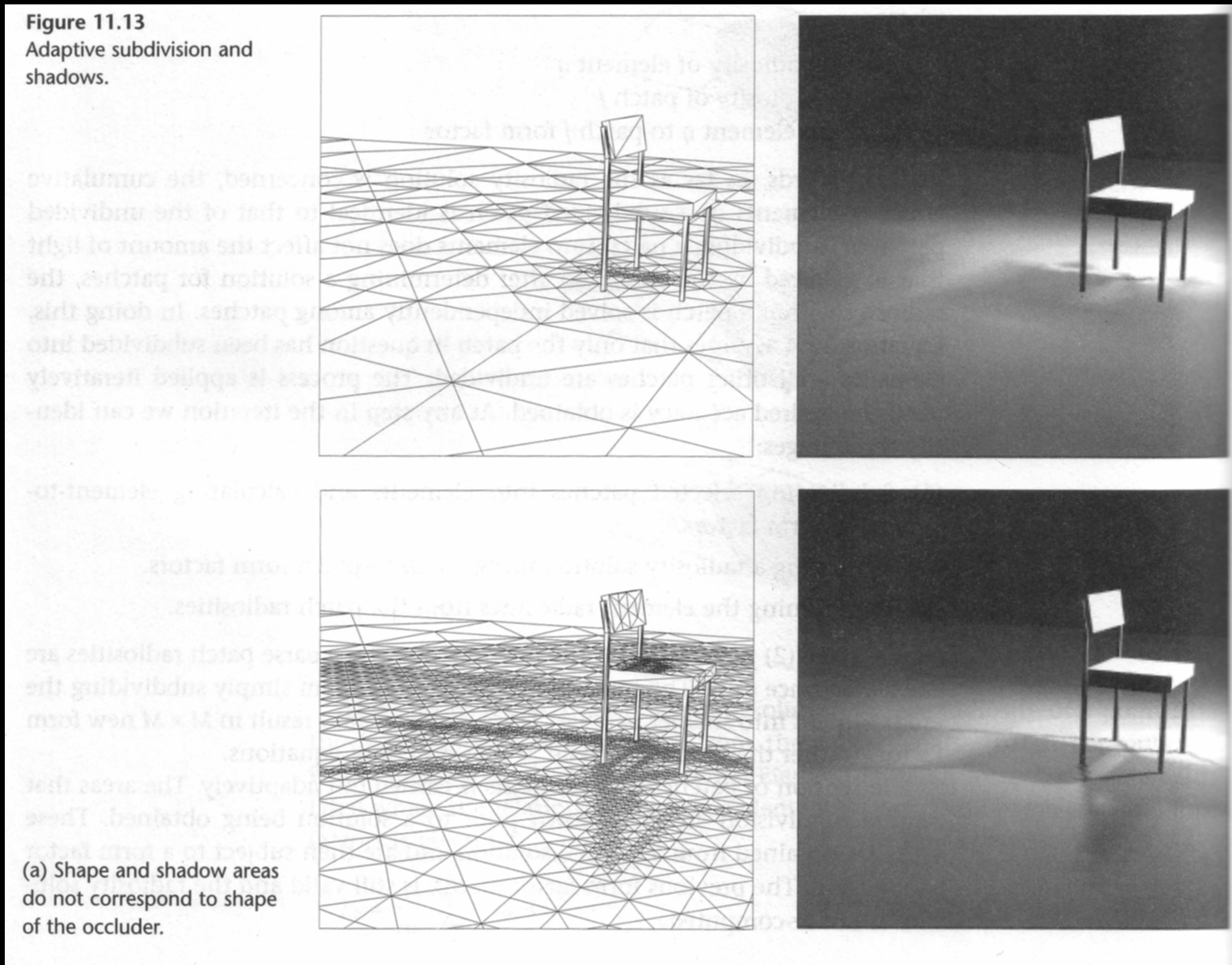
# Substructuring

- Radiosity assumed constant across patch
- Impact of number of patches
  - Few: fast, but very inaccurate (blocky)
  - Many: slow $O(n^2)$, but much more accurate
- Substructuring
  - Introduce elements as a substructure for patches
  - Use adaptively where radiosity varies rapidly
  - Distinguish elements and patches to avoid explosion

# Elements vs. Patches

- Analyse transport from patch onto elements
- Do not analyze element-to-element detail
- This means
  - Compute form factors from elements to patches
  - Do not compute form factors from patches to elements
  - Use weighted patch to parent-of-element
  - Complexity $O(m \cdot n)$ for m elements, n patches
- Typically substructured areas
  - Near lights
  - Shadow boundaries

# Adaptive subdivision and shadows




Watt, 3D CG, 3rd Ed, 2000

# Outline

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

# Matrix Radiosity Revisited

- Compute all form factors $F_{i\,j}$
- Make initial approximation to radiosity
  - Emitting elements $B_i = E_i$
  - Other elements $B_i = 0$
- Apply equation to get next approximation

$$B_i' = E_i + \rho_i \sum_j F_{ij} B_j$$

- Iterate with new approximation
- Intuitively
  - Gather incoming light for each element i
  - Base new estimate *B'* on previous estimate *B*

# Progressive Refinement

- 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

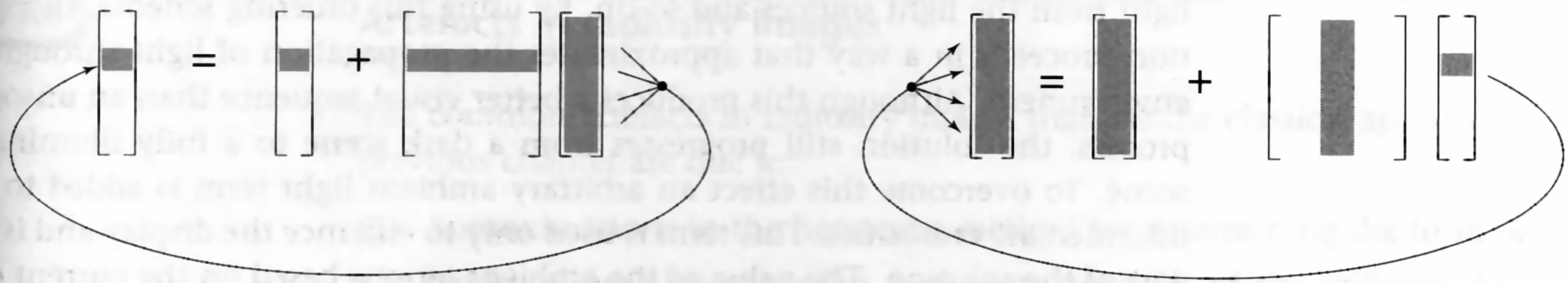
Gathering: a single iteration ($k$) updates a single patch $i$ by gathering contributions from all other patches.

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

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

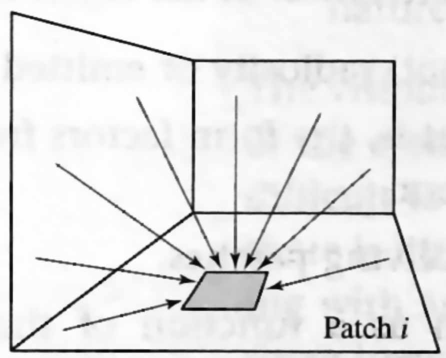
for all $j$:

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

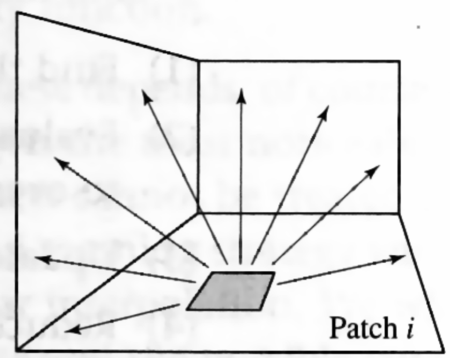



Equivalent to gathering light energy from all the patches in the scene.

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

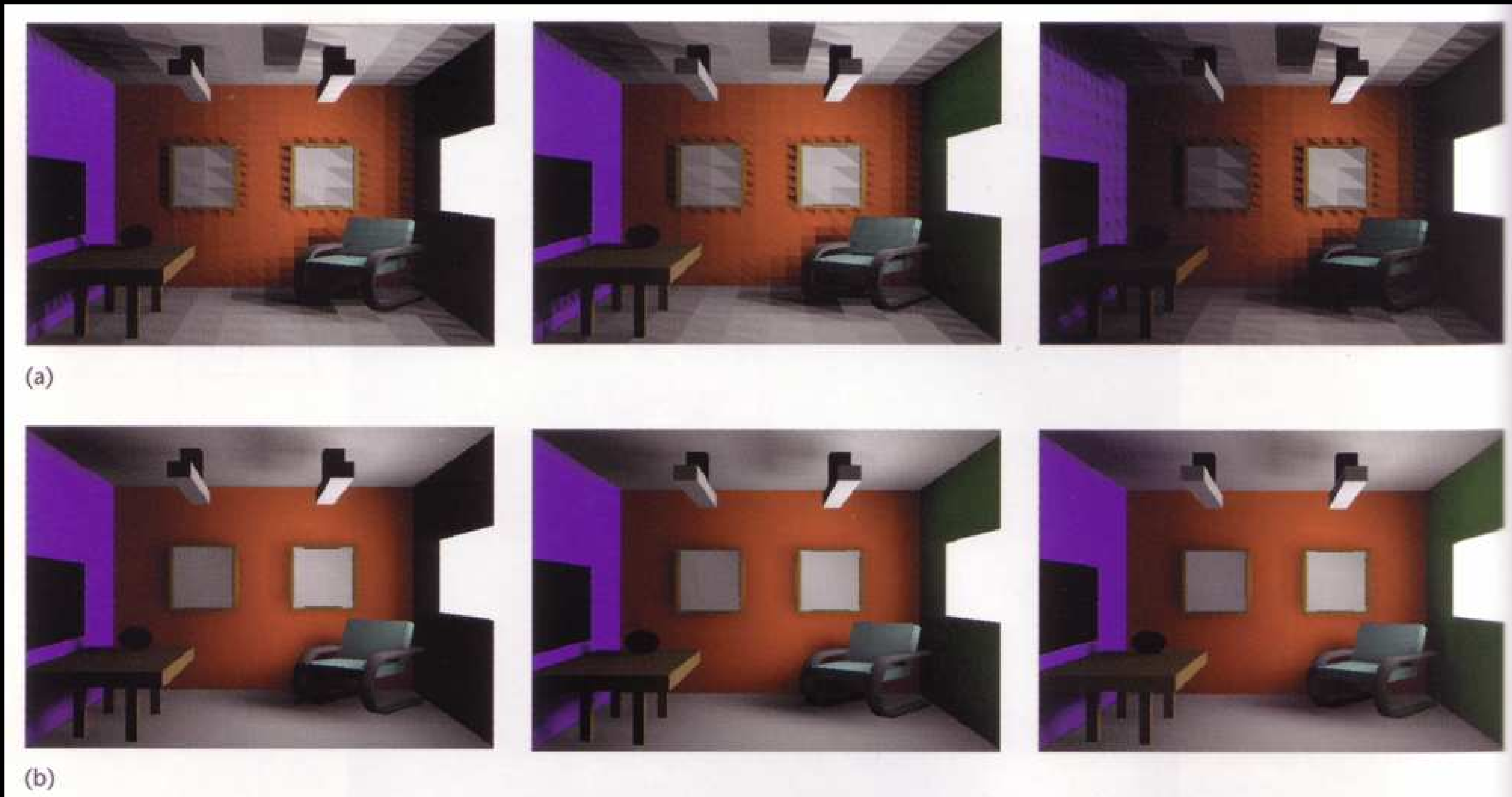
# Progressive Refinement

- Basic algorithm
  - Initialize emitting element with $B_i = E_i$
  - Initialize others with with $B_i = 0$
  - Pick source i (start with brightest)
  - Using hemicube around source, calculate $F_{i\,j}$
  - For each $j \neq i$, approximate $B'_j = \rho_j B_i F_{i\,j} (A_i / A_j)$
  - Pick next source i and iterate until convergence
- Each iteration is O(n)
- May or may not keep $F_{i\,j}$ after each iteration

# Progressive Refinement Corrected

- Problem: double-count if source is used more than once as source
- Solution: compute and use difference from last time a patch was used as a source ($\Delta B_i$), i.e., the *unshot radiosity:*
  - Initialize $\Delta B_i$, $B_i = E_i$
  - Pick source i with maximum unshot power
  - Using hemicube, calculate $F_{i\,j}$ for each j
    - $\Delta R = \rho_j \, \Delta B_i \; F_{i\,j} \, (A_i / A_j)$
    - $B_j = B_j + \Delta R$
    - $\Delta B_j = \Delta B_j + \Delta R$
  - $\Delta B_i = 0$

# Progressive Refinement



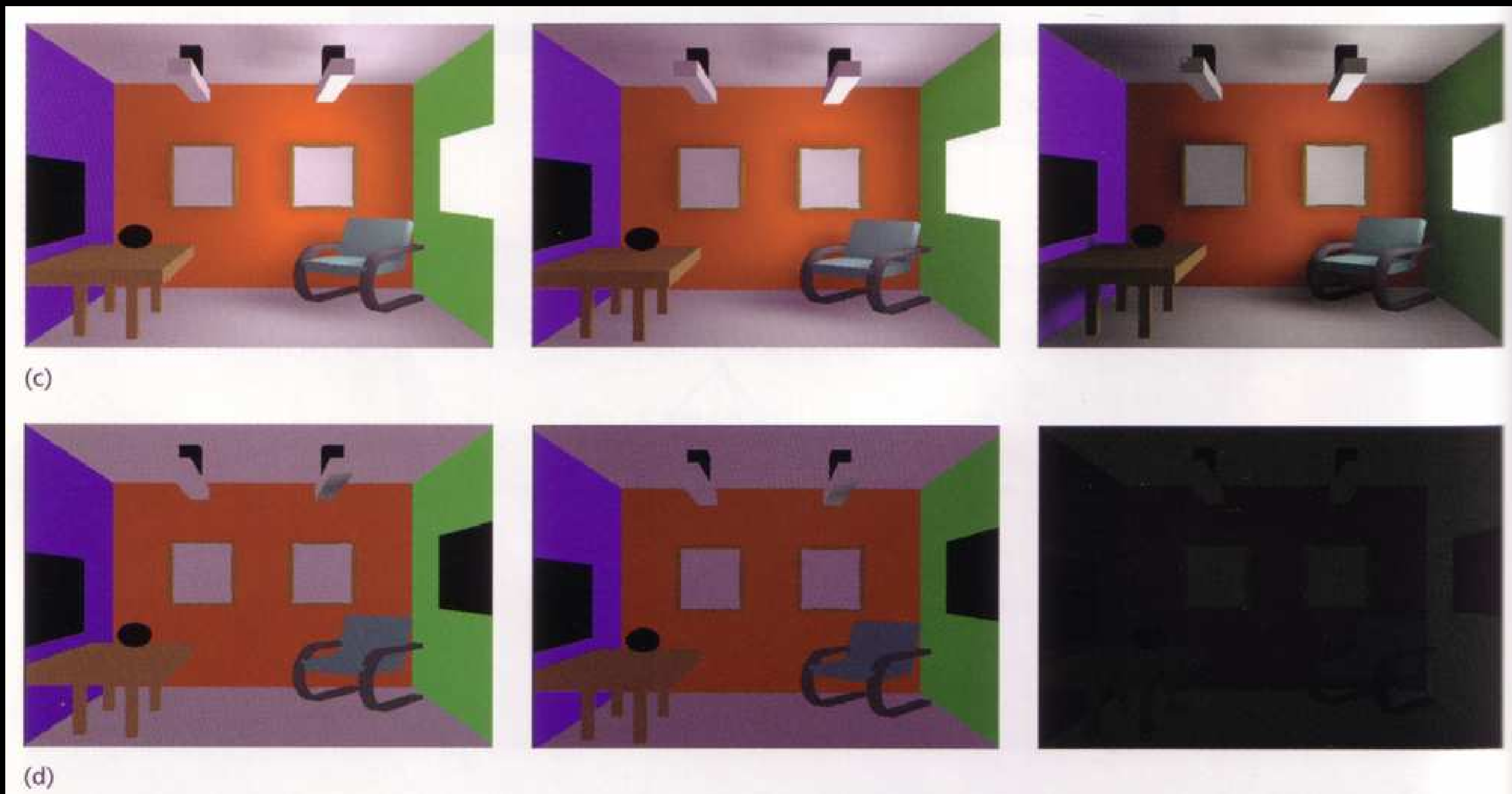

A radiosity image after 20, 250, and 5000 iterations of the progressive refinement method. From top to bottom for each column: (a) The radiosity solution as output from the iteration process. Each patch is allocated a constant radiosity. (b) The previous solution after it has been subjected to the interpolation 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.

# 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

# 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.

# Radiosity Algorithms Summary

- Matrix radiosity algorithm
  - Pre-compute all form factors
  - Iterative solution (Gauss-Seidel)
    - Start with emission
    - Each objects gathers light from all other objects
- Progressive refinement
  - Pick brightest patch
  - Compute outgoing form factors
  - Shoot light from this patch to all other patches
  - Repeat for next brightest batch
- Combine substructuring and progressive refnt.

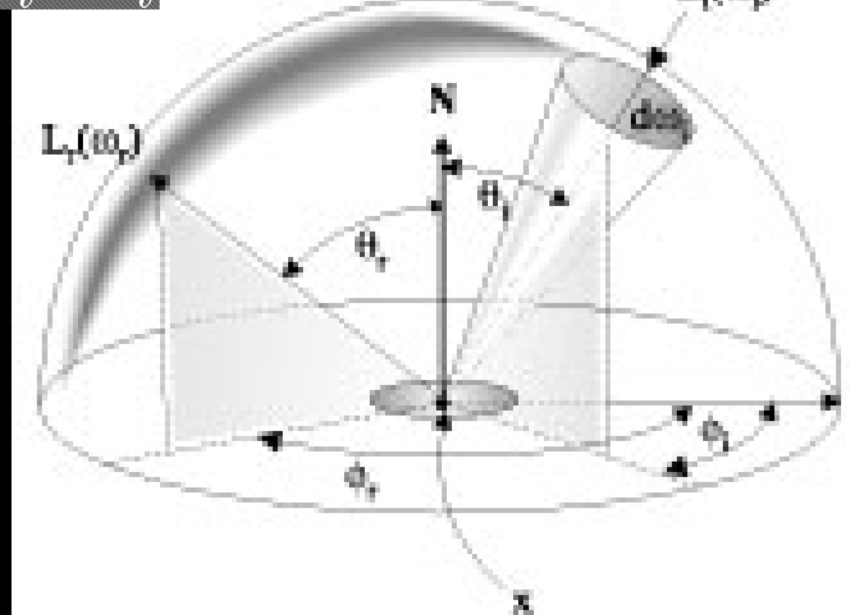
# Outline

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

# Bidirectional Reflectance Distribution

- General model of light reflection
- Hemispherical function
- 6-dimensional (location, 4 angles, wavelength)

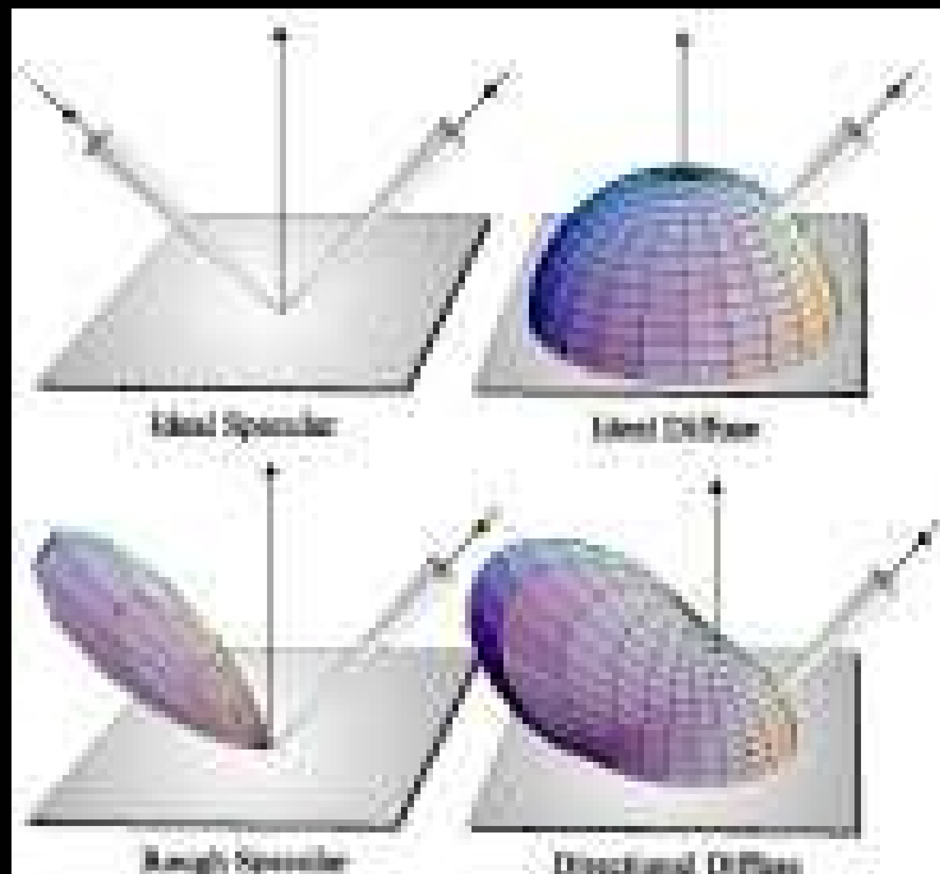
$$f(\omega_i \to \omega_r) = \frac{L_r(\omega_r)}{L_i(\omega_i)\ \cos\theta_i\, d\omega_i}$$



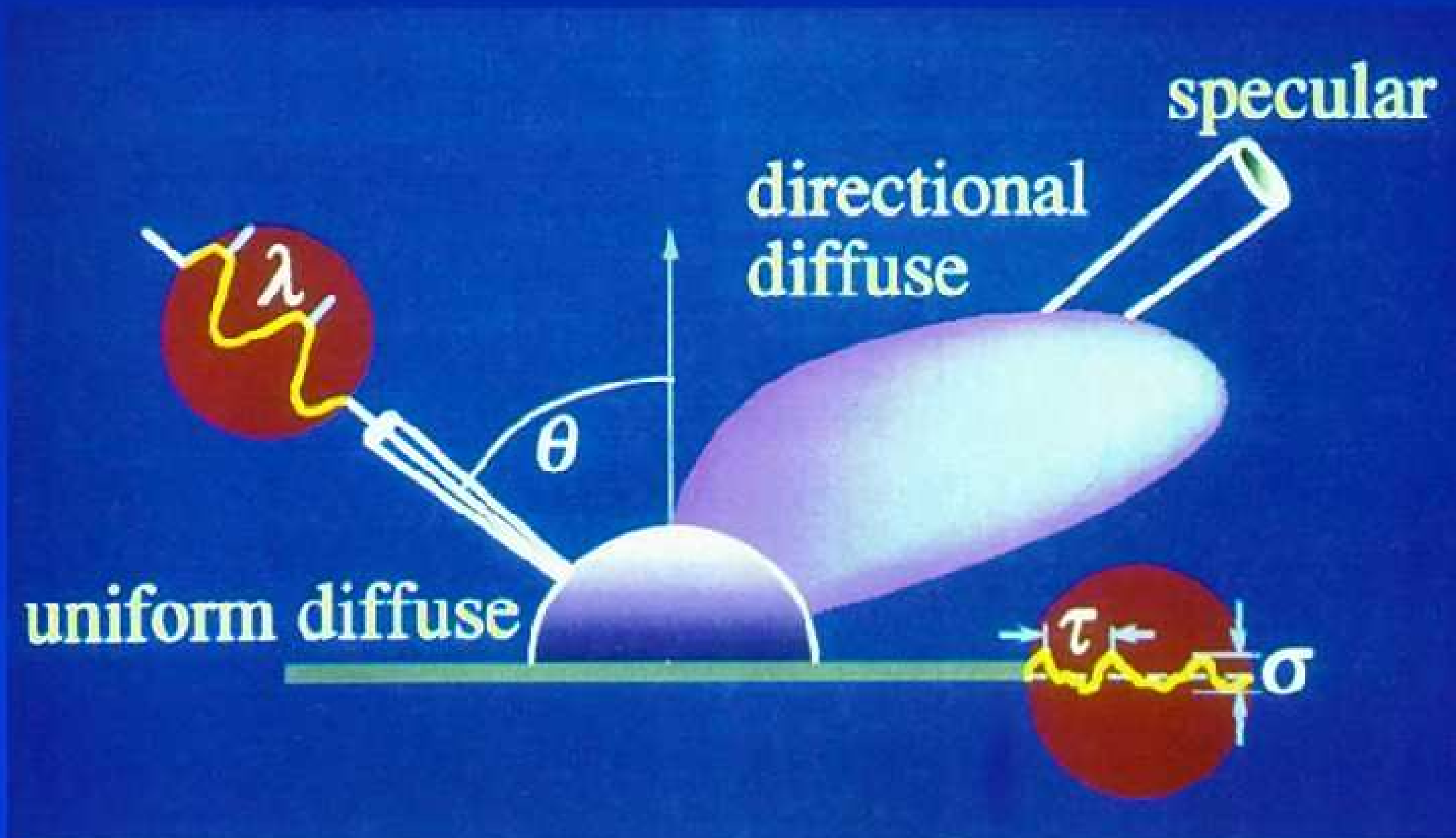

A. Wilkie

# BRDF Examples

- Measure BRDFs for different materials



# Bidirectional Reflectance (BRDF)



# Material Examples

Marschner et al. 2000

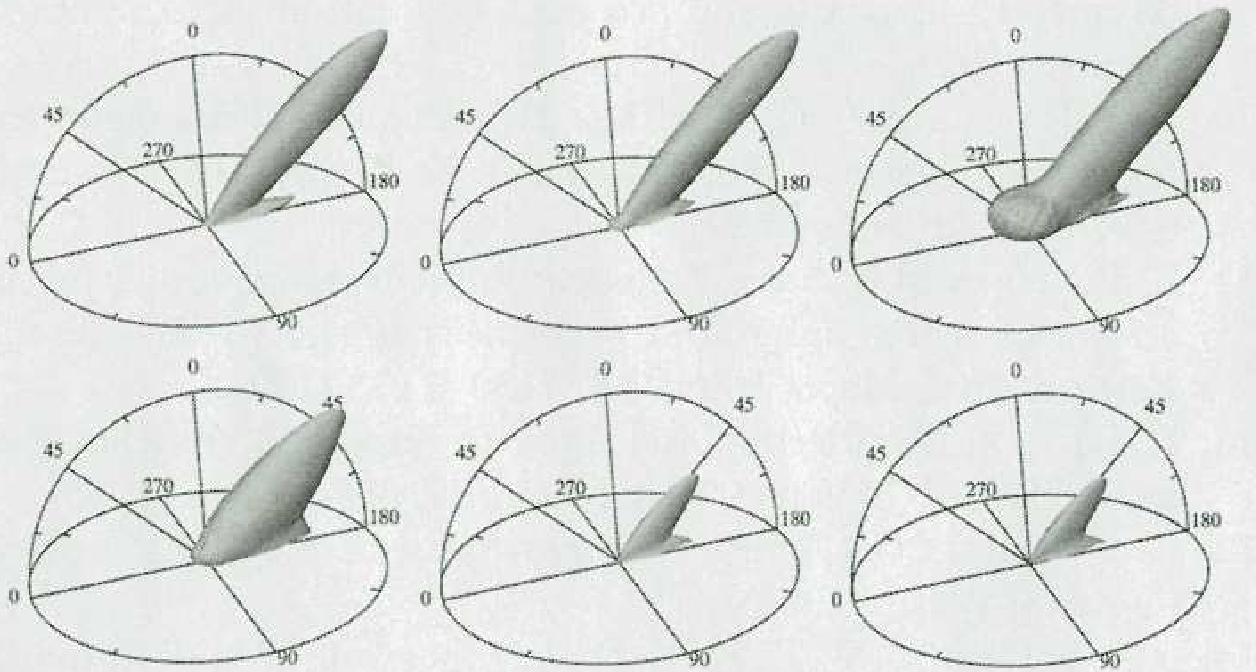



Fig. 16. Resampled scattering diagrams of the BRDF measurements of two paints: a blue enamel (top row) and a red automotive lacquer (bottom row). The RGB color measurements are shown from left to right.

# BRDF Isotropy

- Rotation invariance of BRDF
- Reduces 4 angles to 2
- Holds for a wide variety of surfaces
- Anisotropic materials
  - Brushed metal
  - Others?
- How many parameters for
  - Ideal specular?
  - Ideal diffuse?

# Subsurface Light Transport

- Jensen et al. 2001

Using only BRDF | With subsurface light transport

# Outline

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing

# Light Transport and Global Illumination

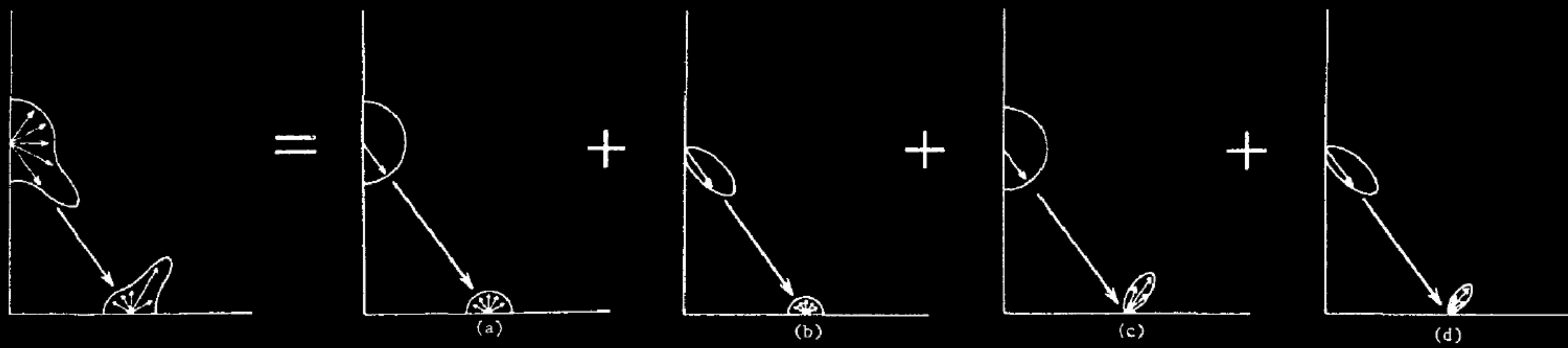



Figure 2. The four "mechanisms" of light transport: (a) diffuse to diffuse, (b) specular to diffuse, (c) diffuse to specular and (d) specular to specular.

- Ray tracing (viewer dependent)
  - Light to diffuse
  - Specular to specular
- Radiosity (viewer independent)
  - Diffuse to diffuse
- Inherent limitations

# Specular Radiosity

- Diffuse radiosity
  - Light reflected equally in all directions
  - Relationship between patches limited to form factor
- Specular radiosity
  - Retain viewer independence (unlike ray tracing)
  - Light reflected differently in different directions
  - For each source and each direction, need to calculation interaction
  - Not practical

# Two-Pass Approach

- A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods, John R. Wallace, Michael F. Cohen & Donald P. Greenberg, SIGGRAPH 87.
- View-dependent specular is tractable
- View-independent diffuse is tractable
- First pass view independent
  – Enhanced radiosity
- Second pass is view dependent
  – Enhanced ray tracing

# Pass 1: Enhanced Radiosity

- Diffuse transmission (translucent surfaces)
  - Backwards diffuse form factor [Rushmeier, 86]
- Specular transmission
  - Extended form factor computation
  - Consider occluding translucent surfaces
  - Window form factor
- Specular reflection
  - Create “virtual” (mirror-image) environment
  - Use specular transmission technique
  - Mirror form factor

# Example: Mirror Form Factors

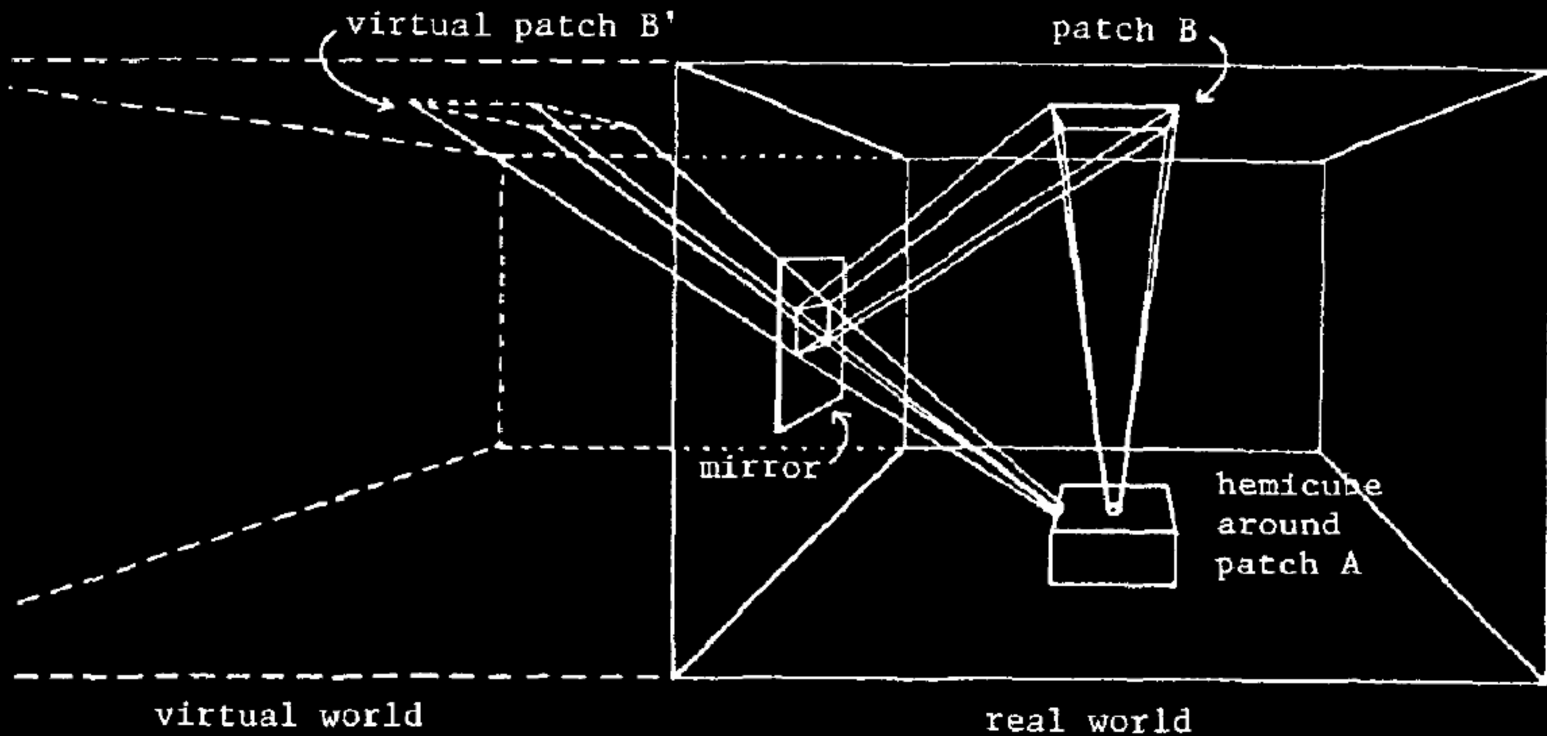



**Figure 3.** Calculation of extra form-factors to account for mirror reflection. Patch A receives light directly from patch B and indirectly through reflection by the mirror. The mirror is treated as a window into a virtual "mirror world." Projecting patch B′ onto the hemicube is then equivalent to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

# Pass 1 Result

- Account only for one specular reflection between surfaces (diffuse-specular-diffuse)
- Accurate diffuse component
- Solve enhanced radiosity equation as before
- Viewer independent solution

# Two-Pass Radiosity Example

(a)

(b)

**Figure 10.** (a) Direct illumination by light sources only.

(b) Diffuse to diffuse transfer included. Specular to diffuse ignored.

(c) Full solution.

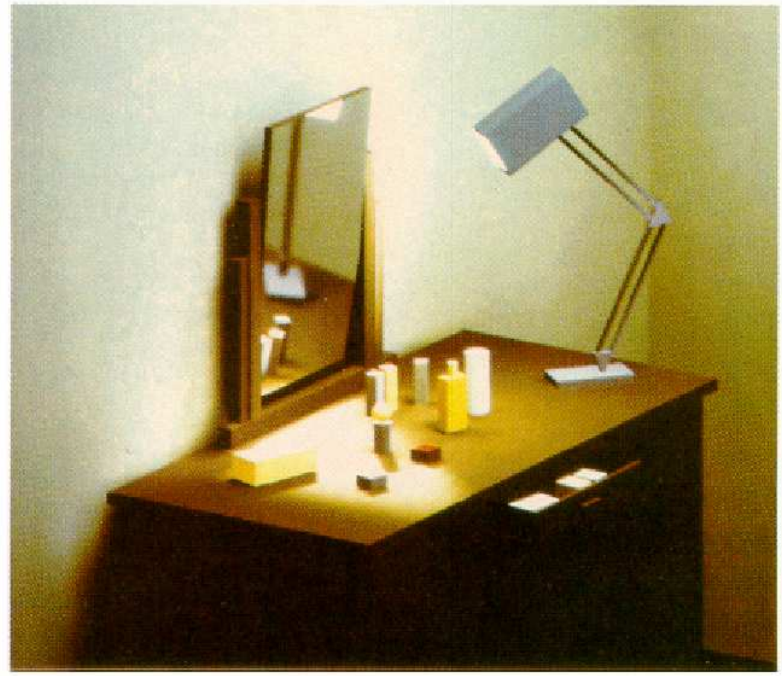
# Pass 2: Enhanced Ray Tracing

- Classical ray tracing
  - Specular to specular light transport
- For diffuse-to-specular transport:
  - Should integrate incoming light over hemisphere
  - Approximate by using small frustum in direction of ideal reflection
  - Use radiosity of pixels calculated in Pass 1
  - Apply recursively if visible surface is specular

# Two-Pass Radiosity Example

(a)

(b)



(c)

**Figure 10.** (a) Direct illumination by light sources only.

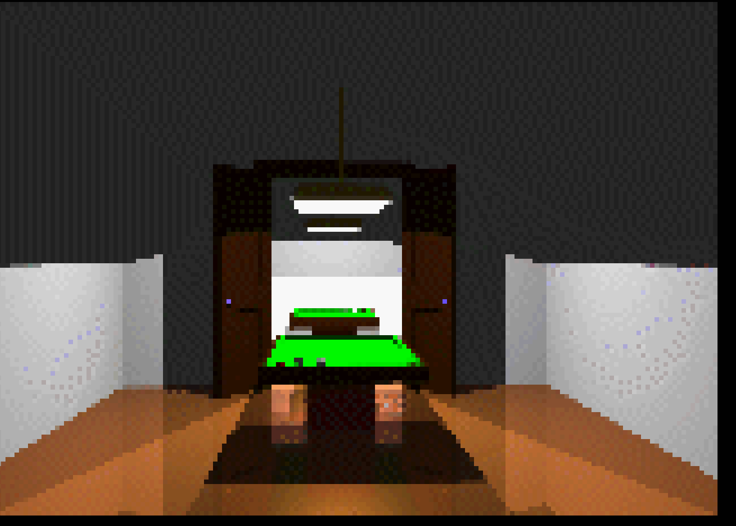
(b) Diffuse to diffuse transfer included. Specular to diffuse ignored.

(c) Full solution.

# Two-Pass Global Illumination

- Still several approximating assumptions
- Appropriate for scenes with few specular reflecting or transmitting surfaces
- More expensive than already expensive methods
- Photon Mapping: Another two-pass algorithm

# Two-Pass Radiosity Example



# Photon Mapping Example

Jensen 1996

# Photon Mapping Example

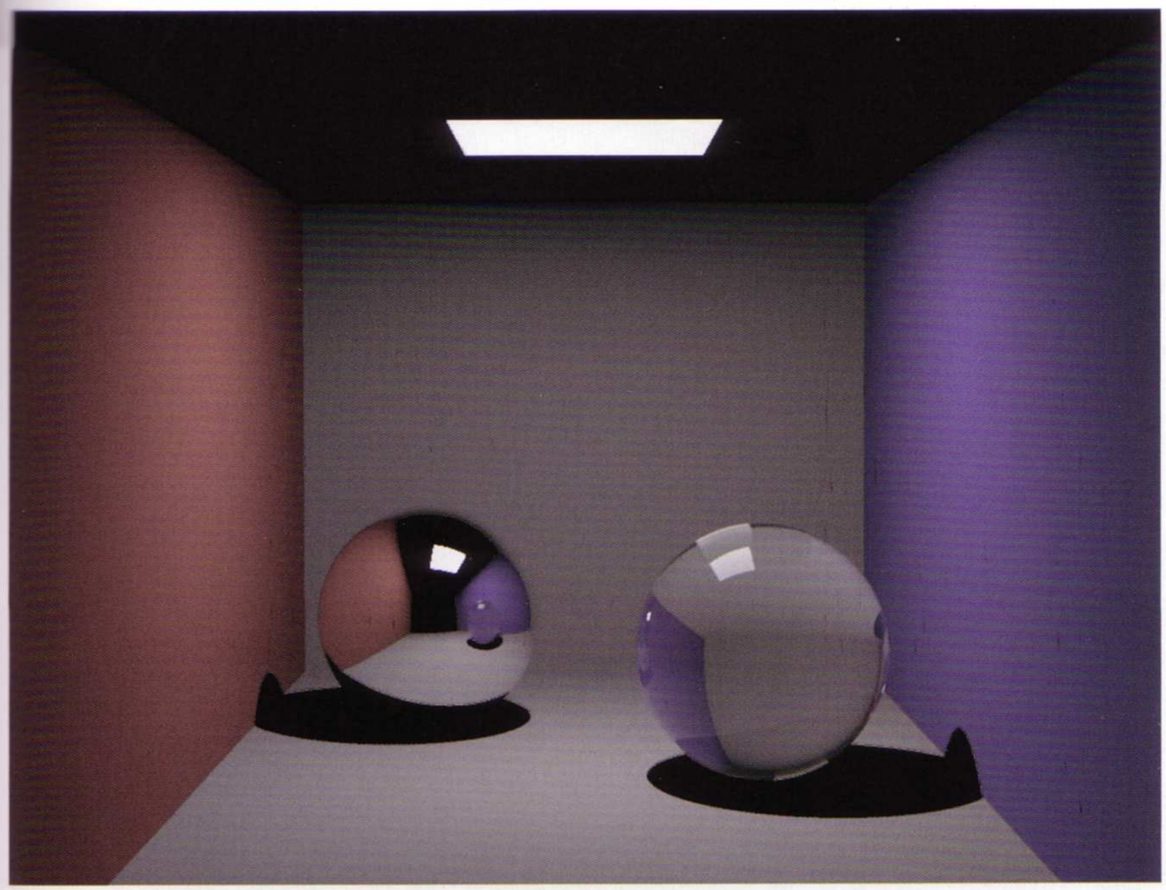


Plate VI. The box scene ray-traced. (See Figure 9.10)

# Photon Mapping Example

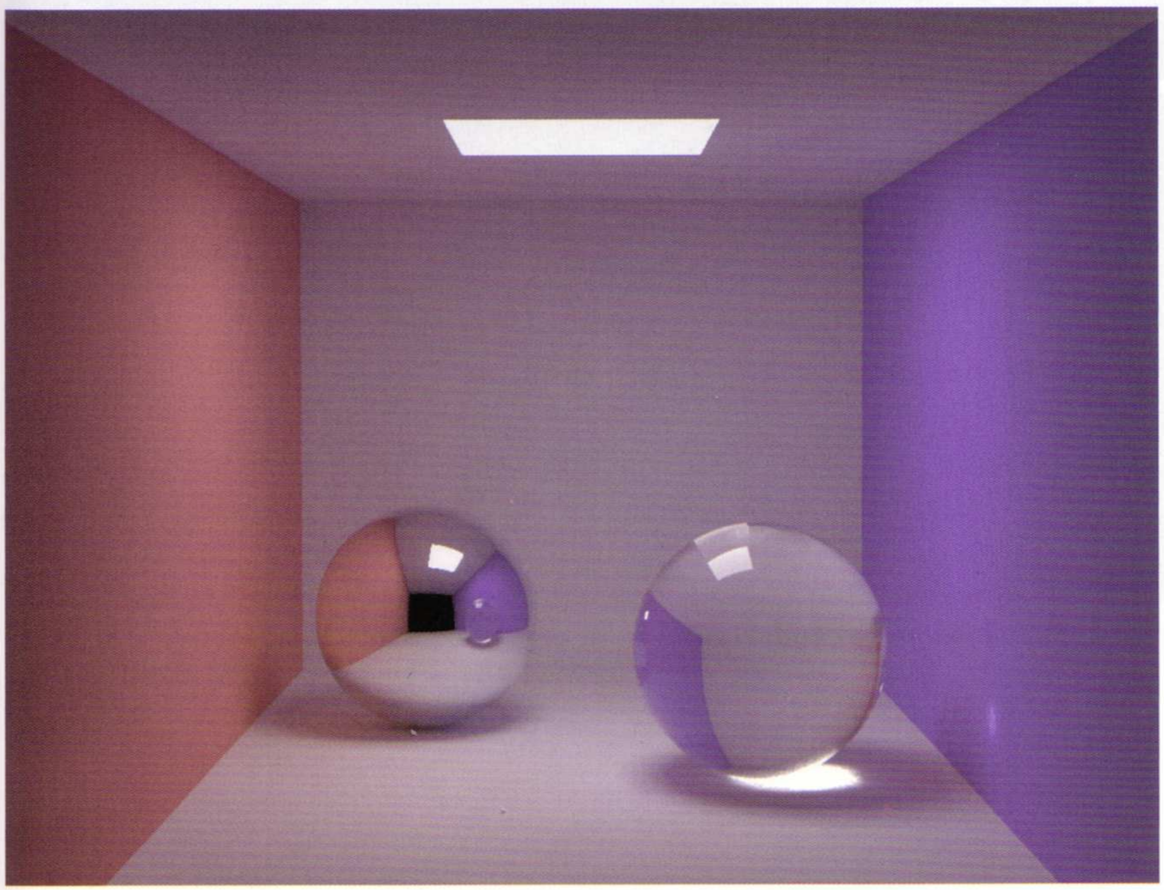


Plate VII. The box scene with full global illumination. (See Figure 9.9)

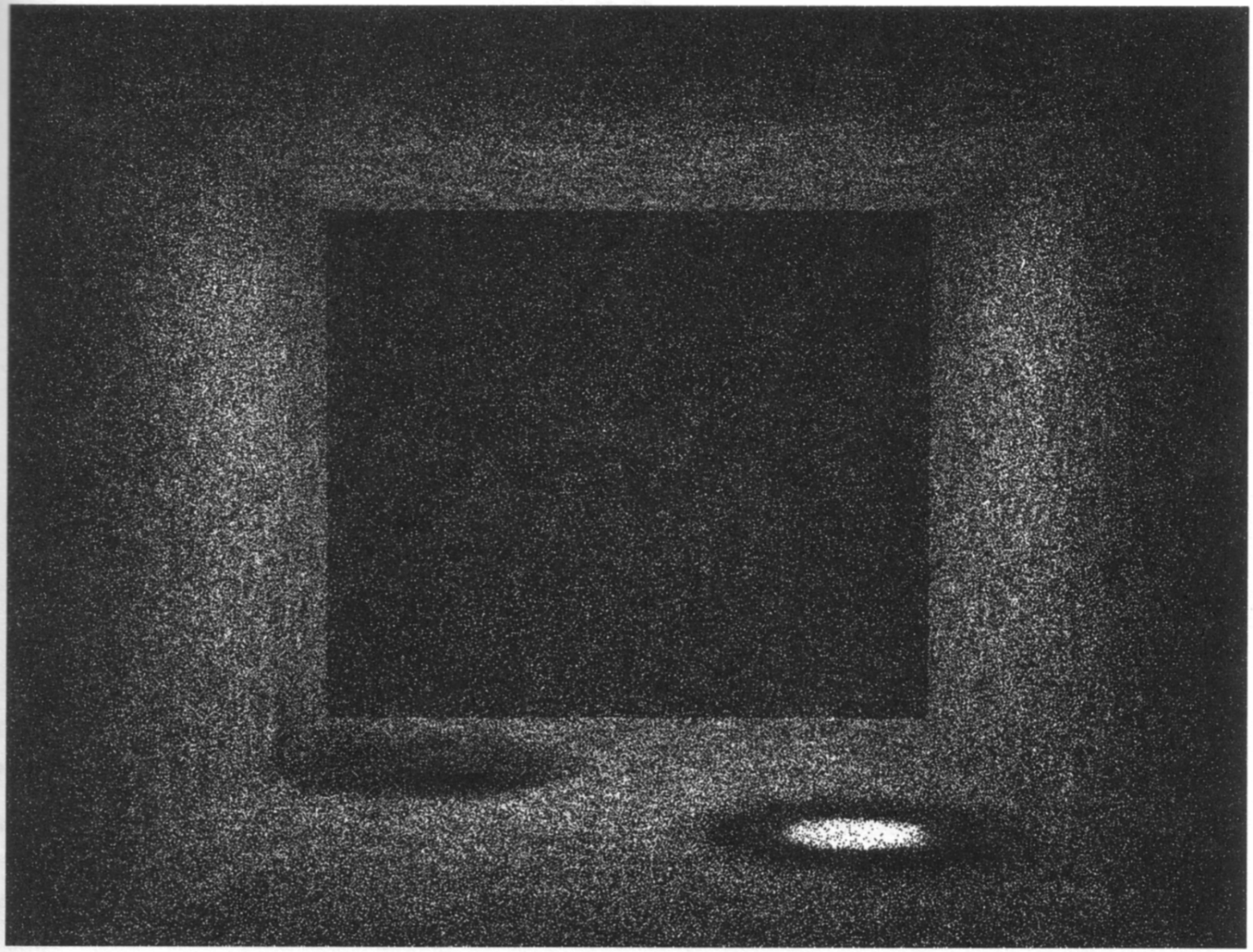


Figure 5.4. The photons stored in the box scene. The top picture shows the box scene, and the lower image shows the photon hits. We used 100,000 photons in this image. The photon hits represent incoming flux in the model. Each photon shows the incoming flux density—the power of the photons multiplied by the local photon density.

# Summary

- Substructuring
- Progressive Refinement
- Bidirectional Reflectance Distribution Function
- Combining Radiosity and Ray Tracing