



Radiosity



Radiosity Concept

Global computation of diffuse interreflections among scene objects

Diffuse lighting changes fairly slowly across a surface

- **Break surfaces up into some number of patches**
- **Assume diffuse illumination constant across each patch**

Diffuse reflection independent of viewing direction

- **Interactive rendering possible**
-



Cornell Box



from Sillion and
Puech, *Radiosity
& Global Illumi-
nation*, 1994.



Steel Mill (55,000 elements)



from Watt and Watt, *Advanced Animation and Rendering Techniques*, 1992.



Radiosity - fundamentals

Radiosity: energy per unit area leaving a surface patch per unit time

Radiosity \times area =

emitted energy + reflected energy

$$B_i dA = E_i dA_i + R_i \int_j B_j F_{ji} dA_j$$

Radiosity will be color of rendered surface

- total energy generated by rendering some number of pixels



Form Factor

Describes geometric relationship between two surface patches

F_{ij} = energy leaving A_i that strikes A_j directly
energy leaving A_i over entire hemisphere

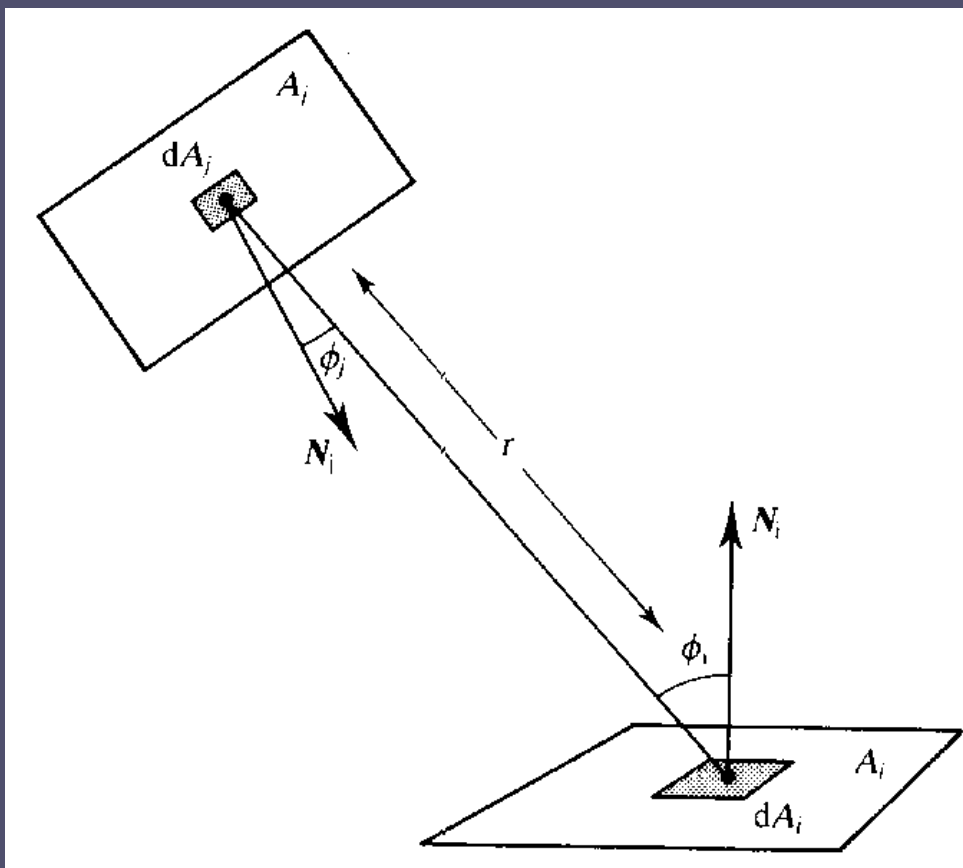
$\sum_i F_{ij} = 1$ for all j

$F_{ii} = 0$ for planar patches

Reciprocity relationship: $F_{ij}dA_i = F_{ji}dA_j$



Form Factor Diagram



from Watt, *3D Computer Graphics*, 1993.

$$F_{ij} = 1/A_i \int_{A_i} \int_{A_j} \cos\phi_i \cos\phi_j / (\pi r^2) dA_j dA_i$$



Differential-Finite Area Form Factor

$$F_{dA_i A_j} = \int_{A_j} \cos \phi_i \cos \phi_j / (\pi r^2) dA_j$$

Form factor between dA_i and A_j

Position dA_i at center of A_i and assume result is valid for entire patch

- reasonable when r is large with respect to areas

Now reasonable to consider projection of patch rather than patch itself...



Patch Projections

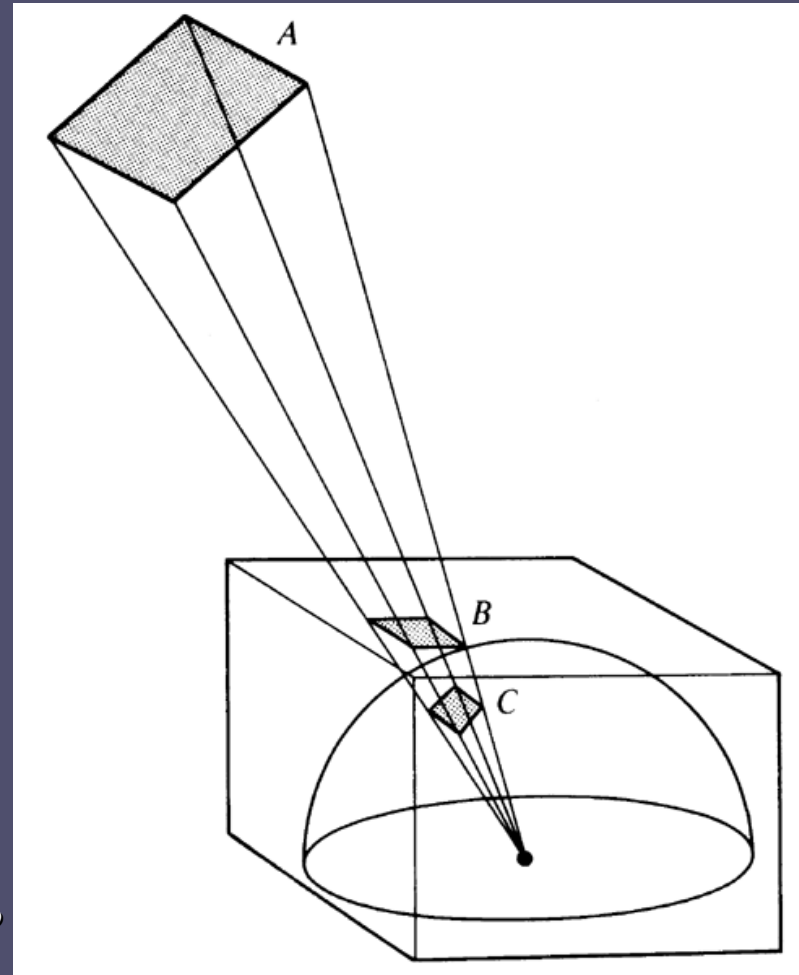
All three representations have the same form factor

A: patch itself

B: patch on hemicube

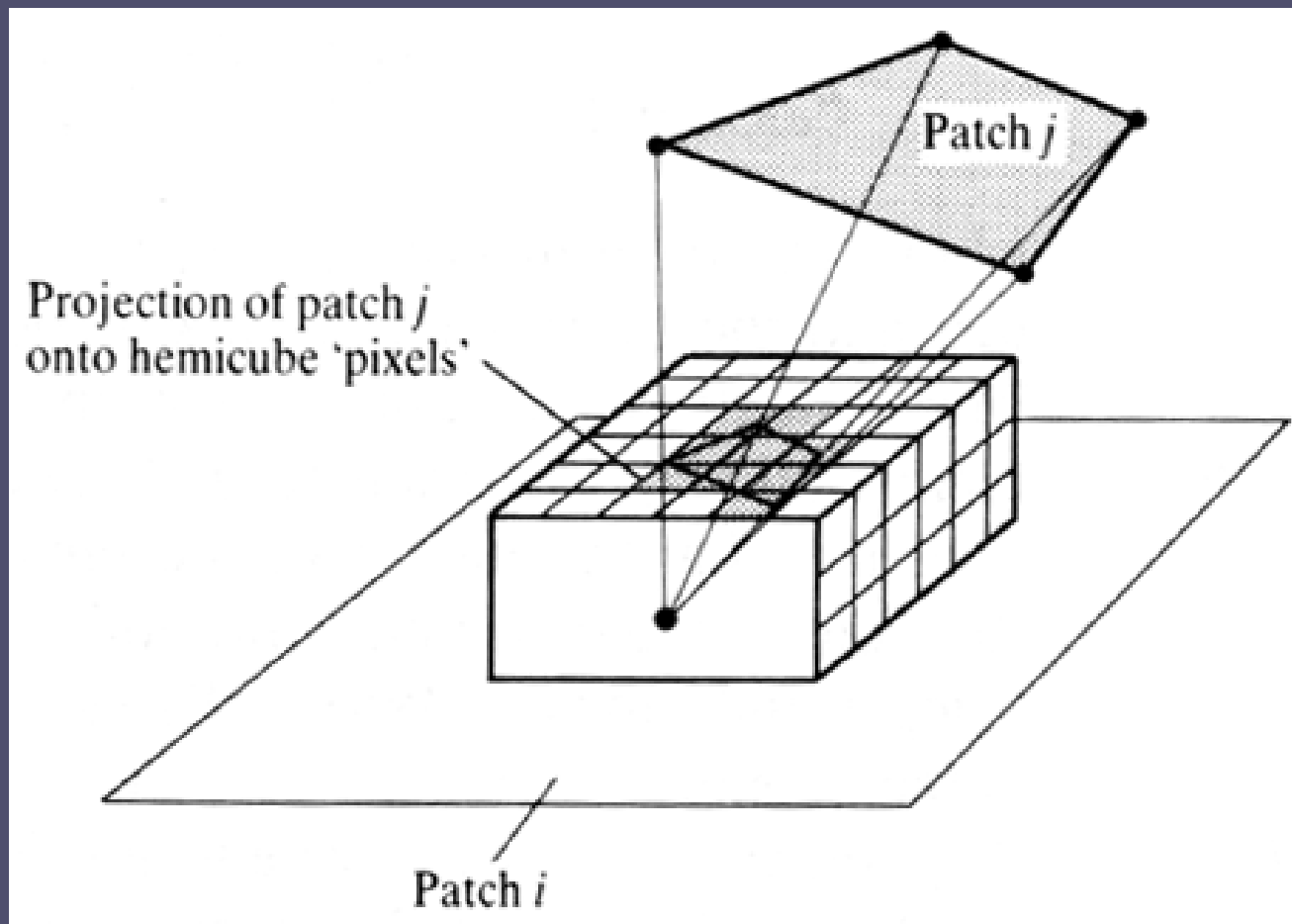
C: patch on hemisphere

from Watt, *3D
Computer Graphics*,
1993.





Hemicube Illustration



from Watt,
*3D
Computer
Graphics,*
1993.



Hemicube Form Factor Method

For each patch, i

For each patch, j

Render patches into item buffer for
each hemicube face (with Z-buffering)

For each hemicube pixel

Look up pixel form factor

Accumulate into form factor for
appropriate patch pair (i,j)



Radiosity Computation

Compute form factors

Solve $N \times N$ matrix equation

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

$$\begin{bmatrix} 1 - R_1 F_{11} & -R_1 F_{12} & \dots & -R_1 F_{1n} \\ -R_2 F_{21} & 1 - R_2 F_{22} & \dots & -R_2 F_{2n} \\ \dots & \dots & \dots & \dots \\ -R_n F_{n1} & -R_n F_{n2} & \dots & 1 - R_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \dots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \dots \\ E_n \end{bmatrix}$$



Gathering Method of Radiosity Computation

Compute form factors

Solve matrix equation using Gauss-Seidel iteration

Solve for one patch radiosity at a time

Plug solution into matrix for solutions to future radiosities

Iterate until it converges



Shooting Method of Radiosity Computation

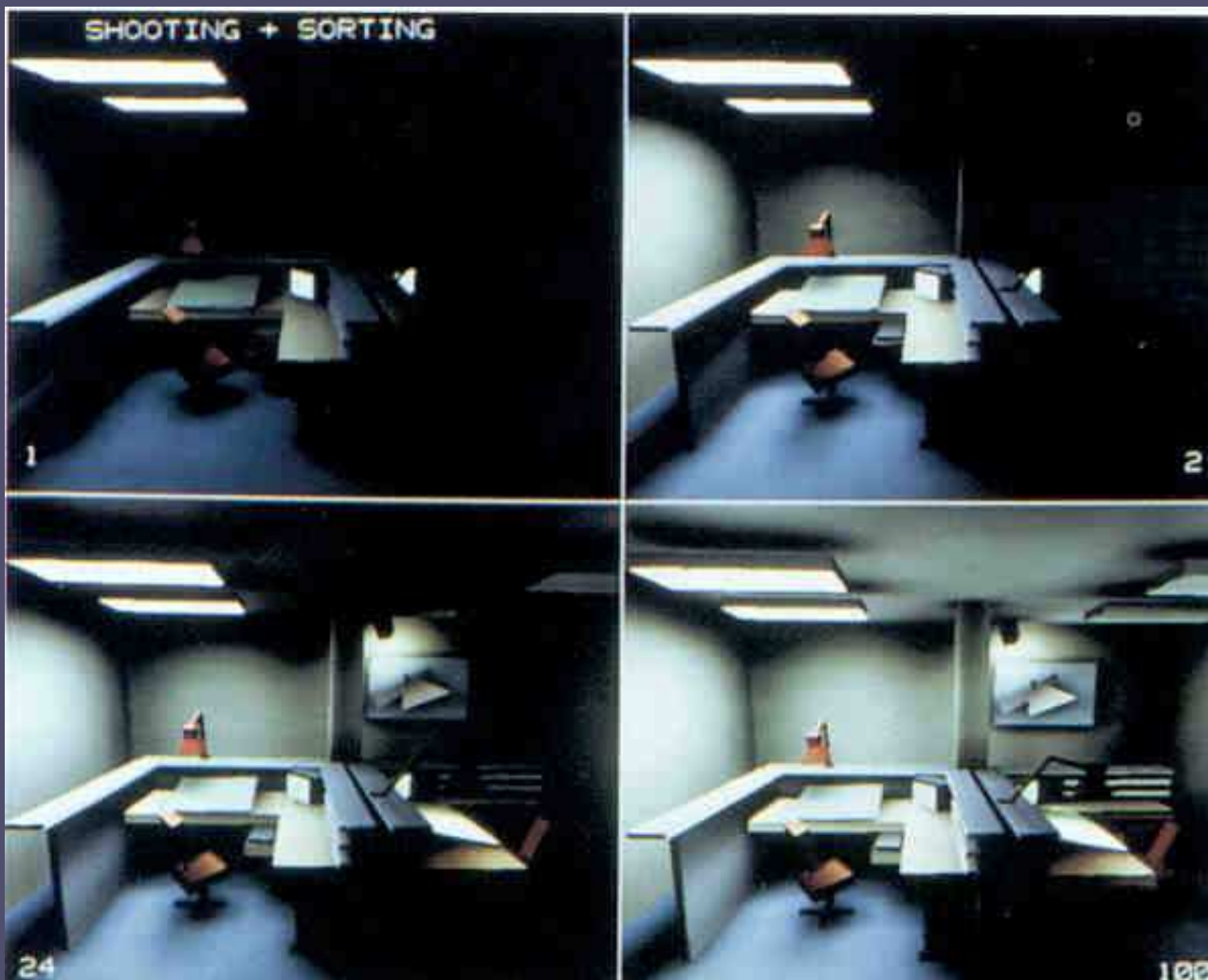
At each iteration, emit from one patch to all other patches

- Useful for progressive radiosity
- Possibly add ambient when viewing preliminary results

Order the patch emissions by magnitude of energy to be emitted



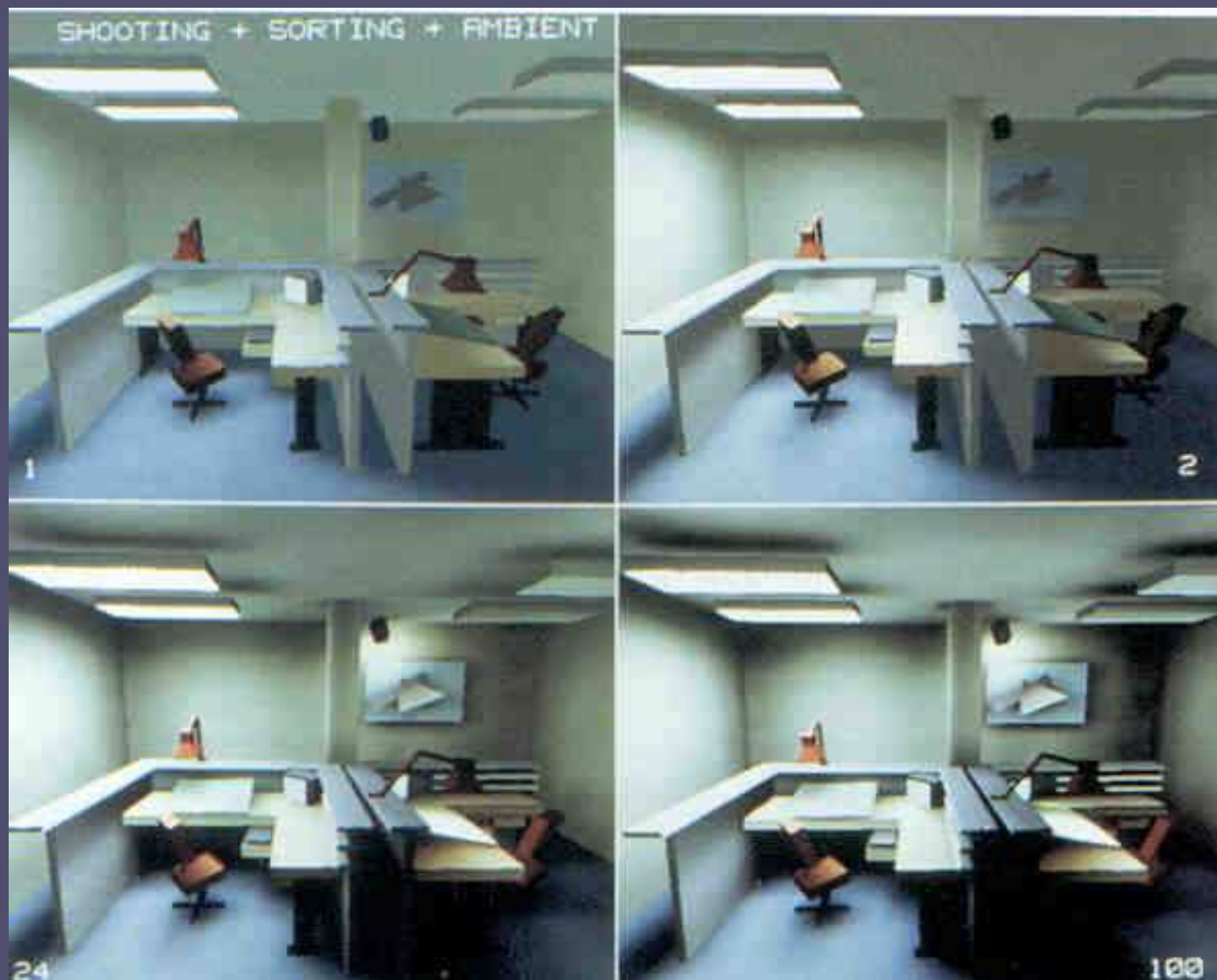
Shooting Example



from Sillion and
Puech, *Radiosity
& Global Illumi-
nation*, 1994.



Shooting + Ambient Example



from Sillion and Puech, *Radiosity & Global Illumination*, 1994.



Creating Patches from Polygons

Uniform subdivision (pre-process)

Adaptive regular subdivision (on-line)

Adaptive irregular subdivision (on-line)



Uniform Subdivision

**Subdivide polygons with regular grid
before any radiosity computation**

**Set some threshold to determine level of
subdivision**

- **number of patches per polygon**
- **maximum patch size**

**Doesn't provide much control in error of
form factor or radiosity computation**



Adaptive Regular Subdivision

Begin with coarse (or no) uniform subdivision of polygons

After computing radiosities, measure gradient between adjacent patches (using differences)

Subdivide patches with high gradient

Incrementally update radiosity solution



Reducing Subpatch Computations

Initialize subpatch radiosities from patch radiosity

Compute only subpatch-patch form factors

- not patch-subpatch form factors
- not subpatch-subpatch form factors

Subdivision effectively increases matrix from $N \times N$ to $M \times N$ (but not $M \times M$)



Hierarchical Radiosity

Apply regular subdivision to patches that require refinement

For each patch-patch interaction, use an appropriate level of subdivision

Can be implemented using matrix block operations

- **portions of matrix are computed as block**
 - **bounds on computational error used to determine which computations may be grouped**
-



Irregular Subdivision (Discontinuity Meshing)

Subdivide patches along discontinuities,
rather than regular subdivision

Discontinuities

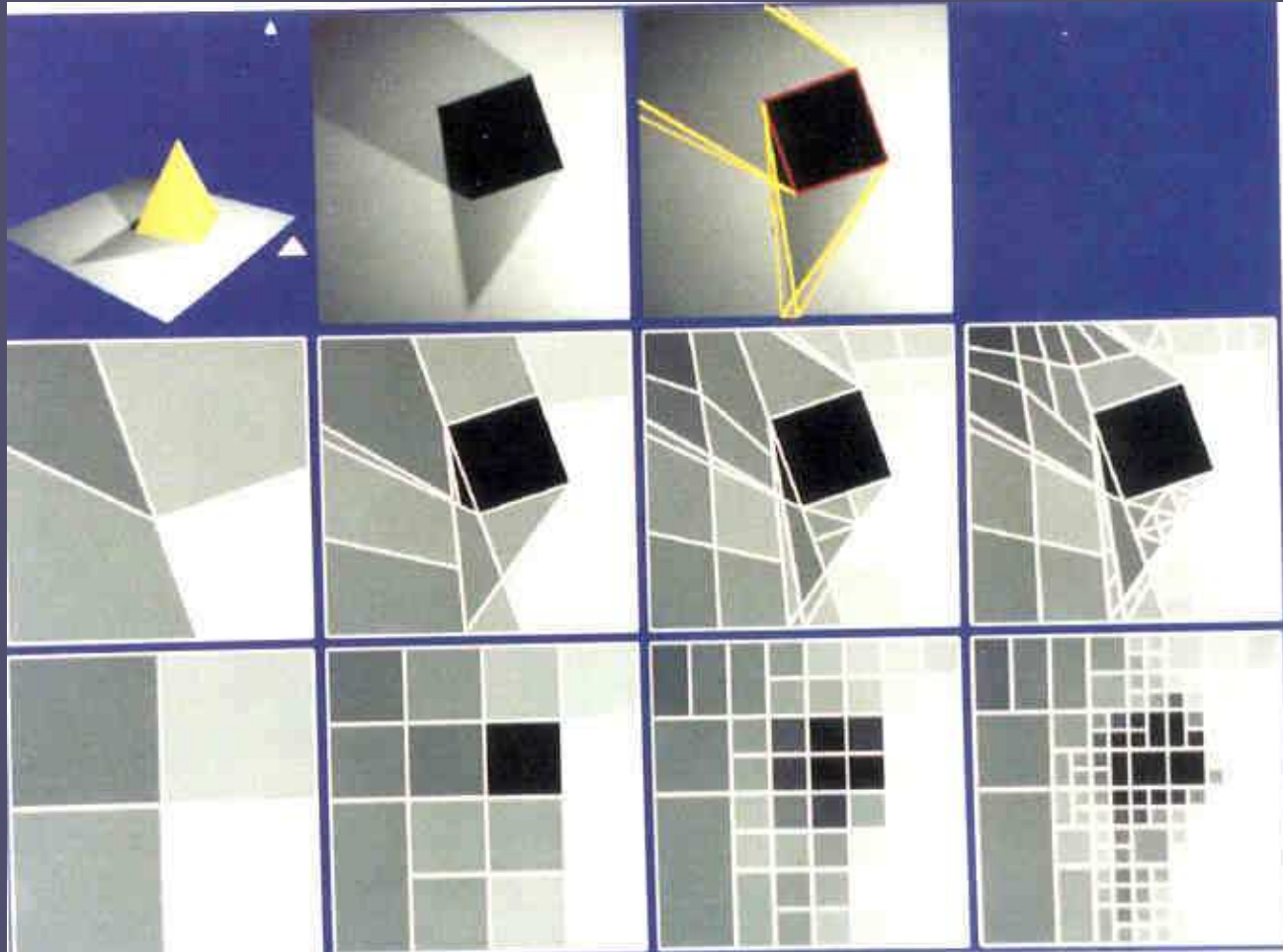
- 0 order: contacts between surfaces
- 1st, 2nd order: changes in visibility

Requires less refinement along
discontinuities than regular subdivision

Typically try to subdivide so most patch
elements completely visible or invisible



Discontinuity Mesh Examples

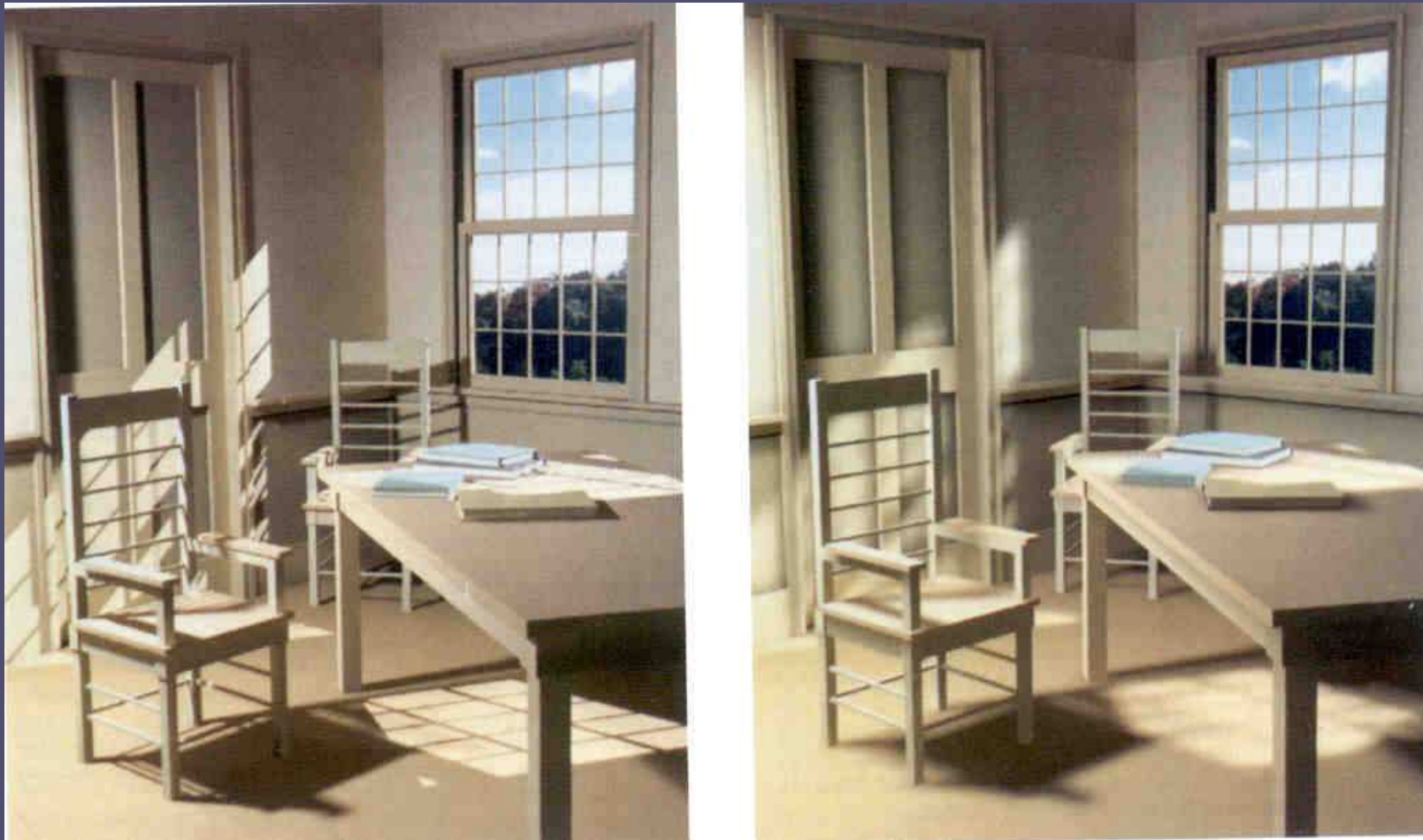


From
Lischinski et
al.,
“Combining
Hierarchical
Radiosity and
Discontinuity
Meshing,”
*Proceedings of
SIGGRAPH
93.*



Discontinuity vs. Regular Subdivision

From Lischinski et al.,
“Combining Hierarchical
Radiosity and Discontinuity
Meshing,” *Proceedings of
SIGGRAPH 93*.





Other Topics of Interest

Combining effects of initial polygons

Using non-constant patch radiositities

**Rendering polygons with higher-order
color interpolation**

Radiosity as textures



Combining Polygon Contributions

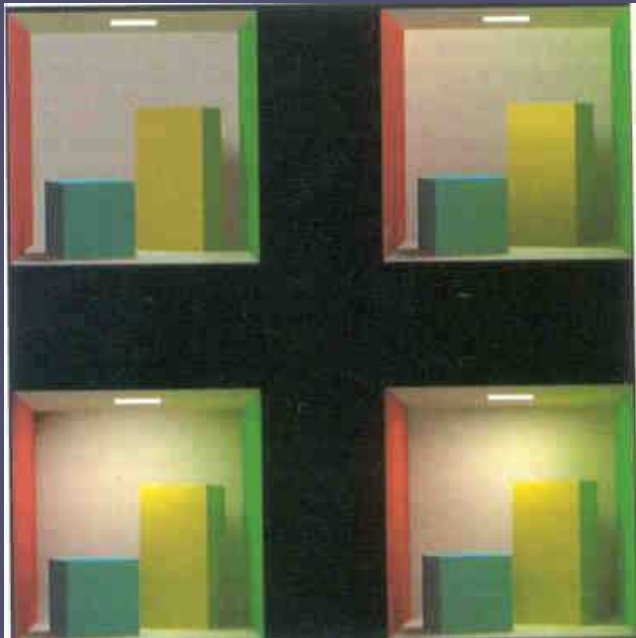
**For polygonal curved surfaces,
simplification allows hierarchical
representation**

**Possibly combine light contributions
through volumes of space**



Non-constant Patch Radiosities

Require fewer patches by allowing radiance to vary across a patch



from Zatz, "Galerkin Radiosity," *Proceedings of SIGGRAPH 93*.



Higher-order Color Interpolation

Using higher-order color interpolation
decreases number of polygons rendered

Higher-order color interpolated polygons
take longer to render

Determine optimum mode for rendering each
patch based on number of polygons and
rendering cost

Explored on Pixel-Planes 5 hardware ~1995



Radiosity as Textures

Accurate radiosity dramatically increases polygon count

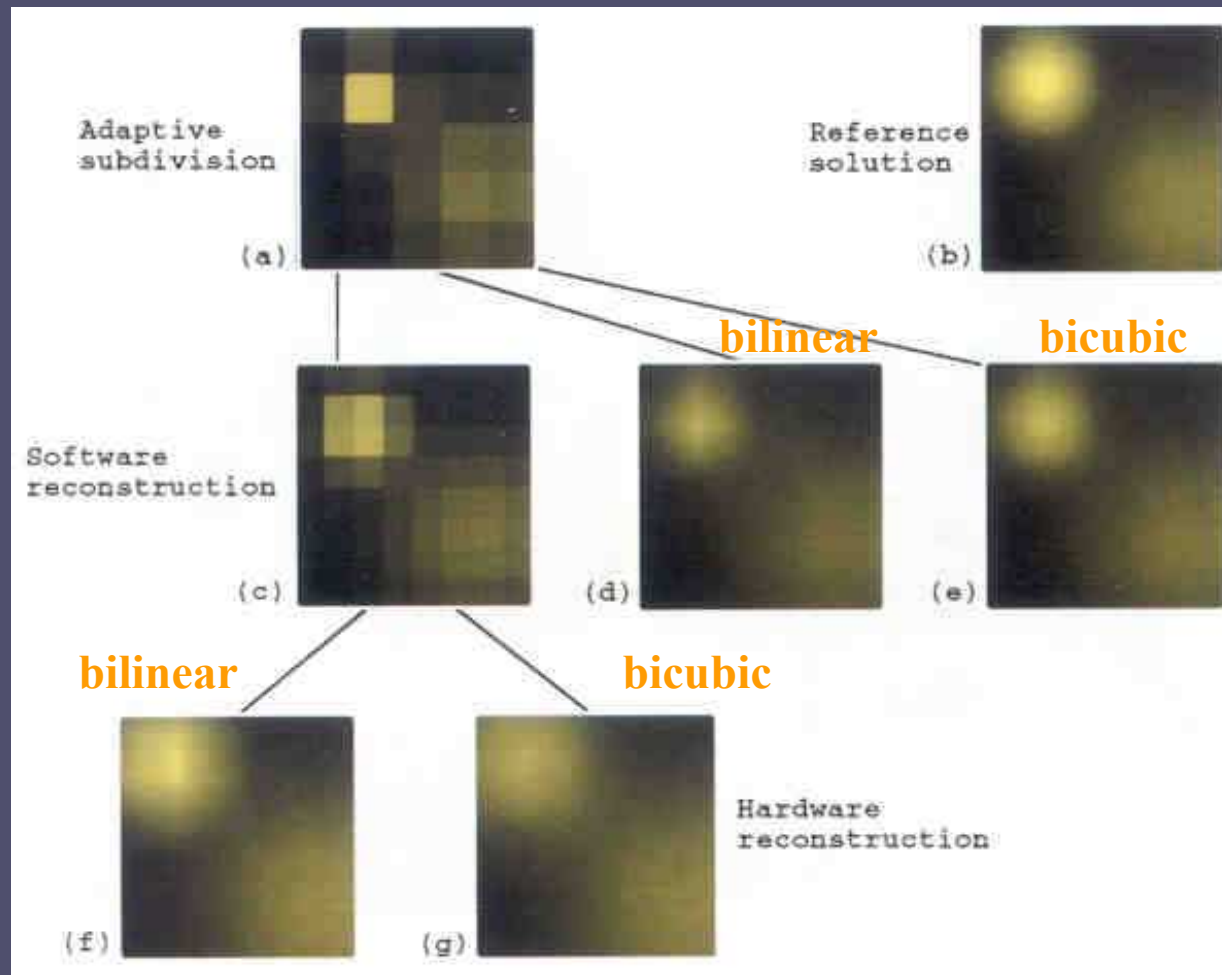
- **Extra geometry is redundant**
- **All new information is about colors**

Create textures for new color information and use original geometry

Like higher-order interpolation, texture-mapping is more expensive than color interpolation, so optimize cost/benefit



Radiosity as Textures Resampling



from Bastos et al.,
“Efficient
Radiosity
Rendering using
Textures and
Bicubic
Reconstruction,”
*Proceedings of the
1997 Symposium
on Interactive 3D
Graphics.*



Video

Bastos, Rui. Michael Goslin, and Hansong Zhang.
**“Efficient Radiosity Rendering using Textures
and Bicubic Reconstruction.”** *Proceedings of the
1997 Symposium on Interactive 3D Graphics.*