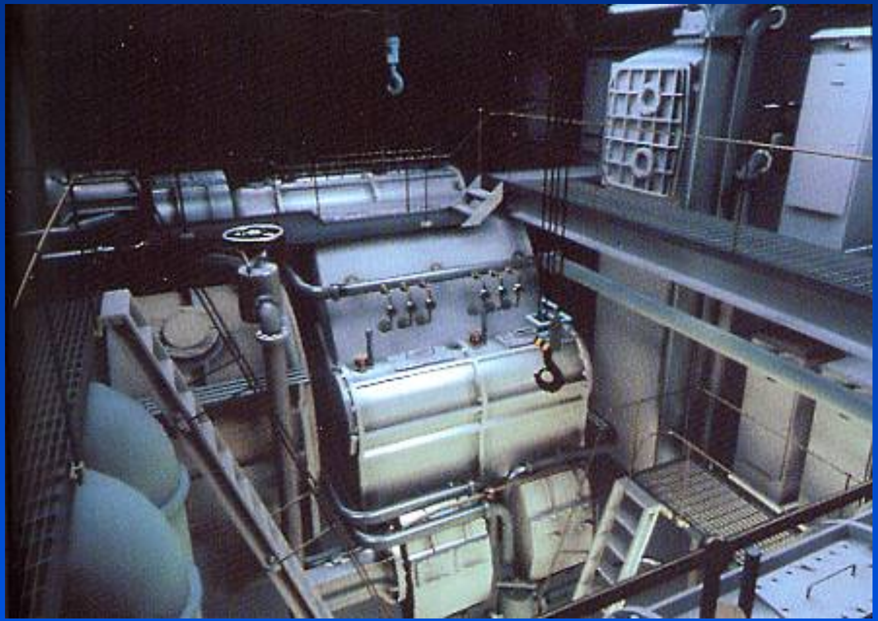
Global Illumination: Radiosity

Radiosity

- Conservation of light energy in closed environment
- Radiosity: rate at which energy leaves surface per unit time, per unit area



“Bunny”
Blue Sky Studios, 1998

For each patch i:

$$B_i = E_i + \rho_i \sum_{1 \le j \le n} B_j F_{j\text{-}i} \frac{A_j}{A_i}$$

B_i, B_j – radiosity of i,j

E_i - rate at which light is emitted by i

ρ_i - i's reflectivity(assume constant Lambertian)

$F_{j\text{-}i}$ - form factor-fraction of energy leaving patch j that arrives at patch i (shape, orientation of patches, obstruction)

A_i, A_j - area of i,j

$B_j F_{j\text{-}i}$ – amount of light leaving unit area A_j reaching all of A_i

multiply by $\frac{A_j}{A_i}$ to get light leaving all of A_j reaching unit area of A_i

- Reciprocity relationship

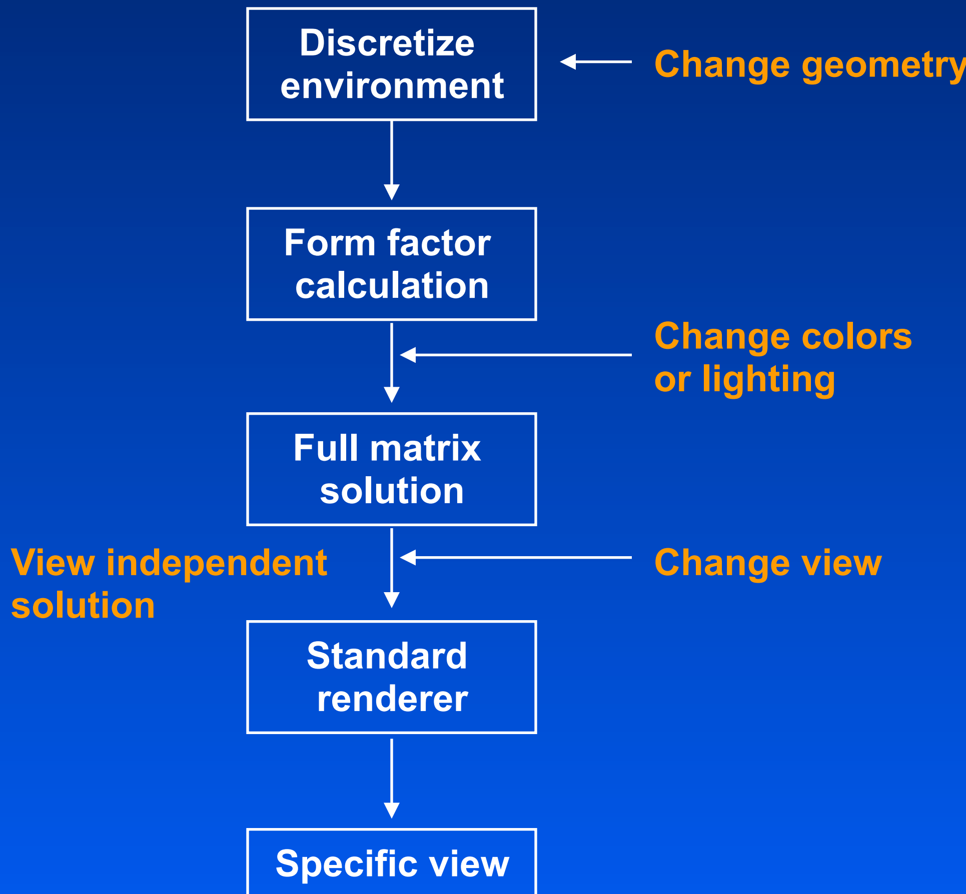
$$A_i F_{i-j} = A_j F_{j-i}$$

$$\Rightarrow B_i = E_i + \rho_i \sum_{i \le j \le n} B_j F_{i-j}$$

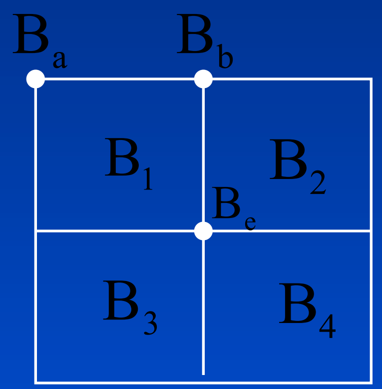
$$\begin{bmatrix} 1-\rho_1 F_{1\text{-}1} & -\rho_1 F_{1\text{-}2} & \cdots & -\rho_1 F_{1\text{-}n} \\ -\rho_2 F_{2\text{-}1} & 1-\rho_2 F_{2\text{-}2} & \cdots & -\rho_2 F_{2\text{-}n} \\ \vdots & \vdots & & \vdots \\ -\rho_n F_{n\text{-}1} & -\rho_n F_{n\text{-}2} & \cdots & 1-\rho_n F_{n\text{-}n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

- Known: ρ, F, E; solve for B at particular wavelengths
- Gauss-Seidel iteration
- For plane or convex patches, $F_{i\text{-}i} = 0$
- Gouraud renderer used for view-dependent solution

Radiosity rendering pipeline



- Calculating vertex radiosities from patch radiosities



$$B_e = \frac{(B_1 + B_2 + B_3 + B_4)}{4}$$

$$\frac{B_b + B_e}{2} = \frac{B_1 + B_2}{2} \Rightarrow B_b = B_1 + B_2 - B_e$$

$$\frac{B_a + B_e}{2} = B_1 \Rightarrow B_a = 2B_1 - B_e$$

Substructuring

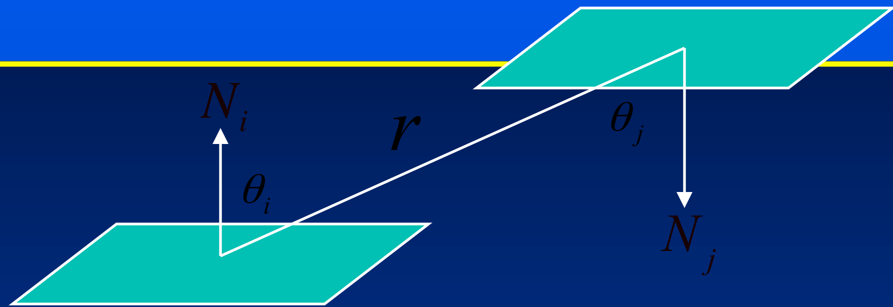
- Subdivide patches into subpatches when high radiosity gradient
  - e.g. at shadow boundary (under objects)
- For each subpatch s, calculate $F_{s\text{-}j}$ for each j
- Replace $F_{i\text{-}j}$ with area-weighted average form factor of m subpatches (i.e. other patches see the large patch i)

$$F_{i\text{-}j} = \frac{1}{A_i} \sum_{1 \le s \le m} F_{s\text{-}j} A_s$$

- Calculate B_i as before, i.e. size of matrix does not change
- For B_s :

$$B_s = E_i + \rho_i \sum_{j=1}^{n} B_j F_{s-j}$$

- Iterate until gradient across subpatch acceptable
- Subpatch's contribution to other patches approximated by large patch's (that contain all the subpatches) radiosity
  - second-order effect



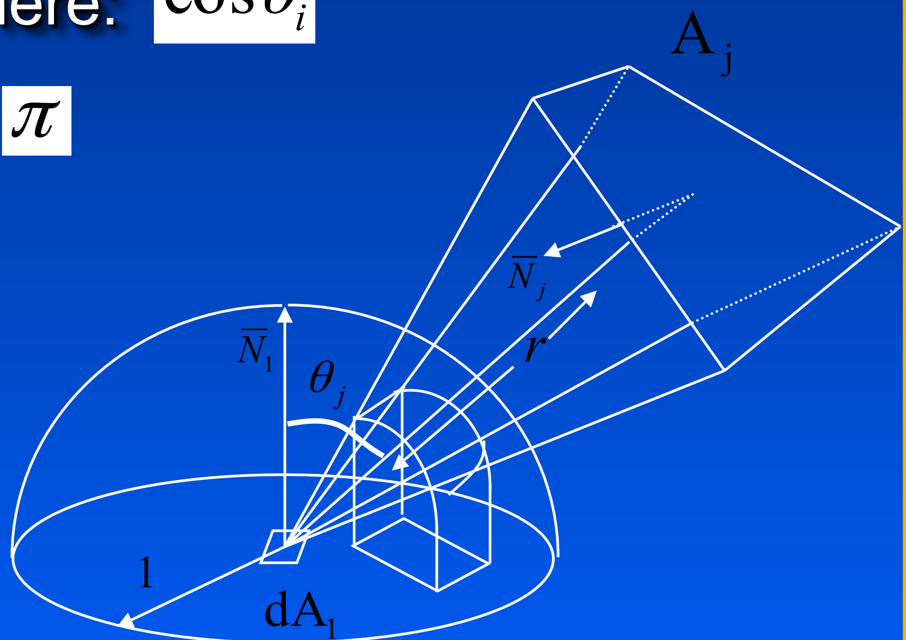
Form Factors

- $$dF_{d_i-d_j} = \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij} dA_j$$ form factor from differential area in i to differential area in j
- $$F_{d_i-j} = \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij} dA_j$$ form factor from differential area of i to finite area in j
- $$F_{i\text{-}j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} H_{ij} dA_j dA_i$$ form factor from A_i to A_j is the area average of above over patch i

- $F_{i\text{-}j} \approx F_{d_i - j}$ if center of patch i typifies other parts
- $\sum_{k=1}^{n} F_{i\text{-}k} = 1$
  - $F_{i\text{-}k}$ fraction of total energy leaving A_i
  - sum to 1 since energy leaving A_i must end up in another patch - in an enclosure
- $F_{i\text{-}i} = 0$ for flat patches – can't shine on itself♪

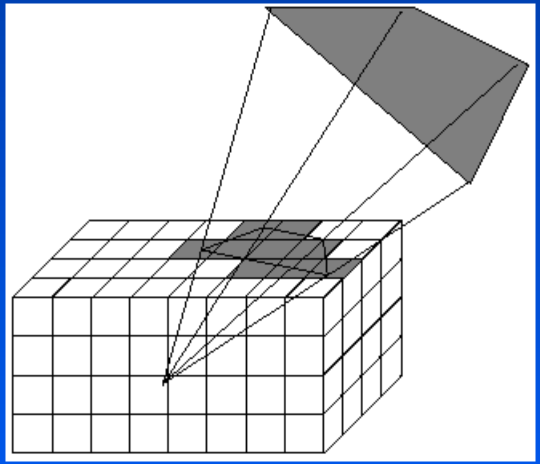
Nusselt's Method

- Project visible part of A_j onto unit hemisphere centered on dA: $\cos\theta_j / r^2$
- Project onto base of hemisphere: $\cos\theta_i$
- Divide by area of unit circle: π



Hemicube Method

- Project visible part of A_j onto hemicube
- Each hemicube cell associated with pre-computed delta form factor value
- Summation of delta form factor values
- Like z-buffer scan conversion



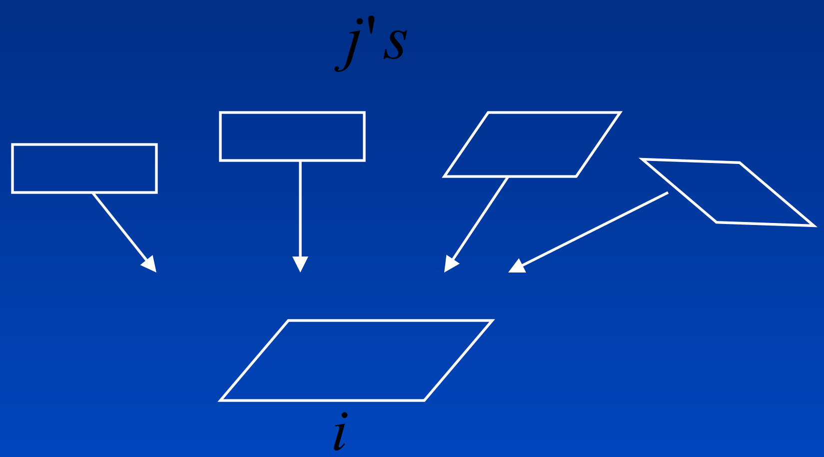
Gauss-Seidel solution (Gathering)

- Initially $B_i = 0 \ \text{(unless } E_i \neq 0)$
- B_i updated using old values of B_j

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ \boxed{B_i} \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ \boxed{E_i} \\ \vdots \\ E_n \end{bmatrix} + \begin{bmatrix} & & & & \\ & & & & \\ \rho_i F_{i-1} & \rho_i F_{i-2} & \cdots & \cdots & \rho_i F_{i-n} \\ & & & & \\ & & & & \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix}$$

- At each iteration, radiosity for one patch calculated by "gathering" radiosities from all other patches
- Entire Gauss-Seidel iteration necessary before interim results are displayable
- Form factors between all patches calculated and stored before solution $O(n^2)$ space

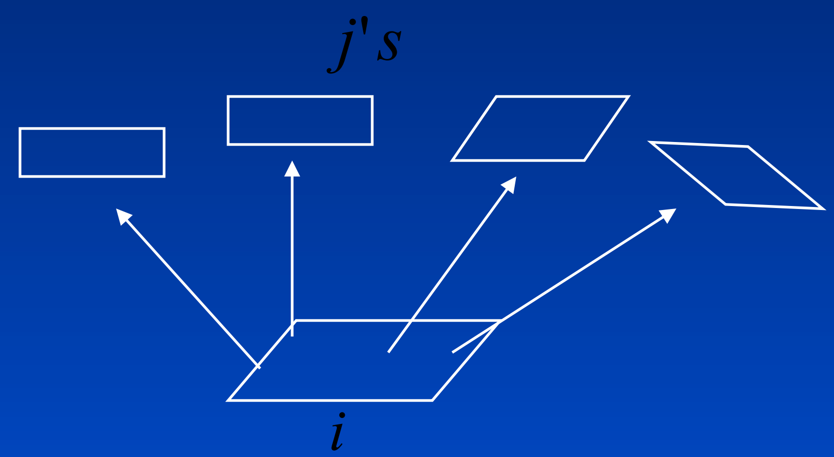
Shooting vs Gathering



B_i due to B_j = $\rho_i B_j F_{i-j}$

for all j

"gathering"



B_j due to B_i = $\rho_j B_i F_{j-i}$

for all j

"shooting"

Progressive Refinement (Shooting)

- Patch i shoot to all other patches

$$
\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ \vdots \\ \vdots \\ B_n \end{bmatrix}
=
\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ \vdots \\ \vdots \\ B_n \end{bmatrix}
+
\begin{bmatrix} \vdots \\ \vdots \\ \vdots \\ \boxed{B_i} \\ \vdots \\ \vdots \end{bmatrix}
\begin{bmatrix} \cdots & \rho_1 F_{1\text{-}i} & \cdots \\ \cdots & \rho_2 F_{2\text{-}i} & \cdots \\ & & \\ \cdots & \rho_n F_{n\text{-}i} & \cdots \end{bmatrix}
$$

Progressive Refinement (Shooting)♪

- Use reciprocity relationship: $$\mathrm{B_j \text{ due to } B_i} = \rho_j \mathrm{B_i F_{i\text{-}j}} \frac{\mathrm{A_i}}{\mathrm{A_j}}$$
  - for all *j* only one hemicube and its form factor need to be stored at a time
- At each iteration shoot $\Delta \mathrm{B_i}$ -radiosity that patch *i* received since last time it shot

- For each patch i (sorted by emission), shoot its unshot energy to every other patch j
  - Initially, it will be the E
- Set its unshot energy to 0 since it just shot its energy

select patch i { /* sort by emission */

calculate $F_{i\text{-}j}$ for each patch j

/* shoot ΔB_i to all other j */

for each patch j {

$\quad \Delta Radiosity = \rho_j \Delta B_i F_{i\text{-}j} \; A_i / A_j$

$\quad \Delta B_j += \Delta Radiosity$

$\quad$ /* ΔB_j contains radiosity accumulated

$\quad$ since last time it shot */

$\quad B_j += \Delta Radiosity$

$\quad$ /* B_j contains total radiosity

$\quad$ accumulated */

}

$\Delta B_i = 0$

temporary ambient term from all ΔB_j

display

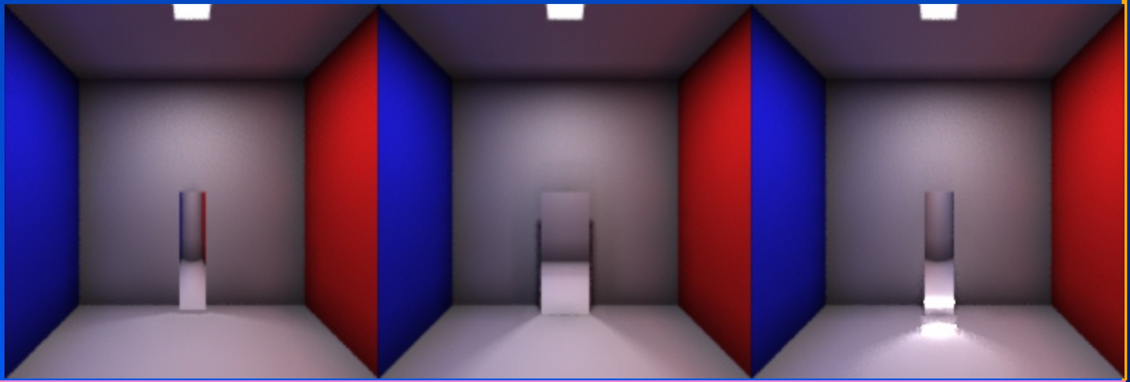
} until convergence

Estimation of temporary Ambient term

- Average reflectivity of entire environment: $$\rho_{avg} = \sum_{i=1}^{n} \rho_i A_i \Big/ \sum_{i=1}^{n} A_i$$
- Overall reflection factor: (number of reflections) $$R = 1 + \rho_{avg} + \rho_{avg}^{2} + \rho_{avg}^{3} + \cdots = \frac{1}{1 - \rho_{avg}}$$
- Approximate form factor fraction of entire environment taken up by *j* $$\mathrm{F}_{\mathrm{i\text{-}j(est.)}} = \frac{A_j}{\sum_{j=1}^{n} A_j}$$
- Ambient term $$I_a = R \sum_{j=1}^{n} \Delta B_j F_{i-j(\mathrm{est.})} \qquad \Delta \mathrm{B}_\mathrm{j} - \text{radiosity of each patch}$$
- Radiosity for display purposes only: $$\mathrm{B}_\mathrm{i}' = \mathrm{B}_\mathrm{i} + \rho_i \mathrm{I}_\mathrm{a}$$

Light transport

- Diffuse to diffuse: radiosity
- Specular to diffuse: ?
- Diffuse to specular: incorrectly by ray-tracing (LOCAL diffuse component passed using specular path)
- Specular to specular: ray-tracing



Extend radiosity to handle Specular terms

- Rather than a single radiosity, discretize hemisphere over patch into finite set of solid angles-each with a radiosity
- Calculate radiosities based on reflectivity (BRDF) that depend on incoming/outgoing direction
- Massive set of equations

Extend Ray tracing to handle diffuse term: Monte Carlo

- Diffuse interreflection with Monte Carlo integration of multiple reflected rays over BRDF
- Very slow rendering speed for many diffuse surfaces

Photon Mapping

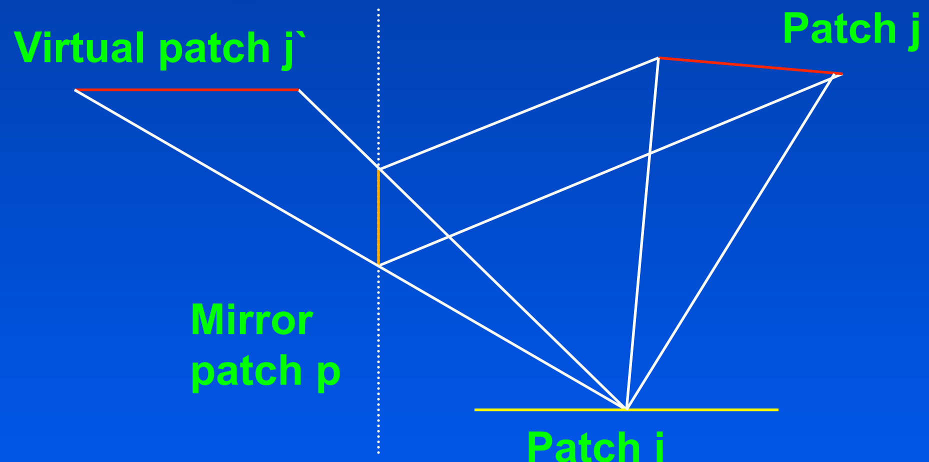
- Light sources shoot photons and reflecting surfaces bounce photons
- Photon is an entity with light energy and direction
- Illumination is calculated by gathering photons absorbed at surfaces
- A view-dependent pass uses photon map to calculate final image
- Must shoot many photons for arbitrary BRDF

Combining Radiosity and Ray Tracing

- View-point independent
  - diffuse-good (Radiosity)
  - specular-difficult
- View-point dependent
  - specular-good (Ray-Tracing)
  - diffuse-difficult

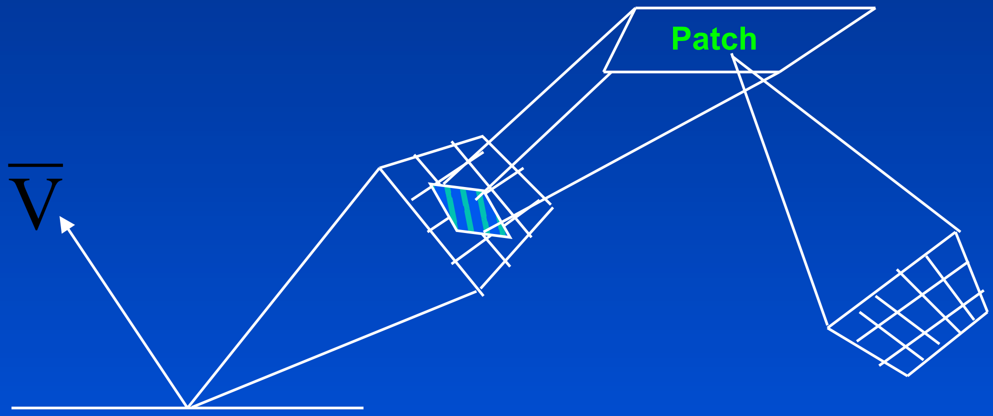
First pass

- Regular radiosity augmented to account for specular-to-diffuse
- Limit to plane mirrors – mirror world
- “mirror” form factor between patch i and j'



Second pass

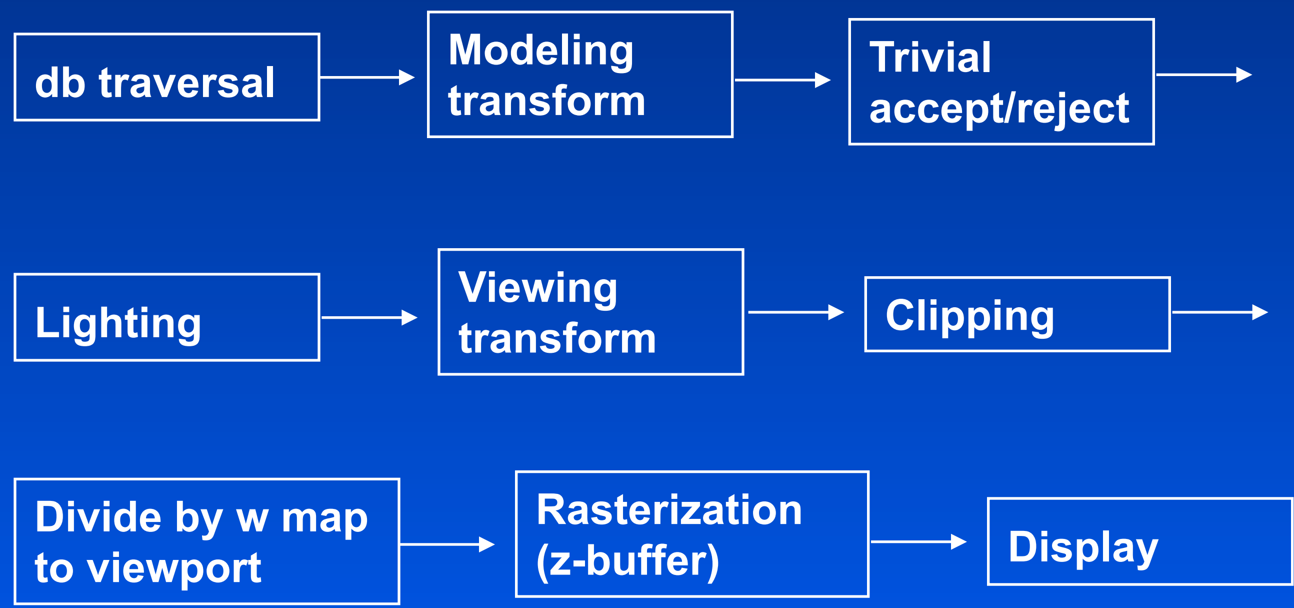
- Reflection frustum(z-buffer) allow weighted sum of specular rays



- Gouraud shading used to interpolate radiosities calculated in first pass

Rendering Pipeline Review

Z-buffer, Gouraud shading

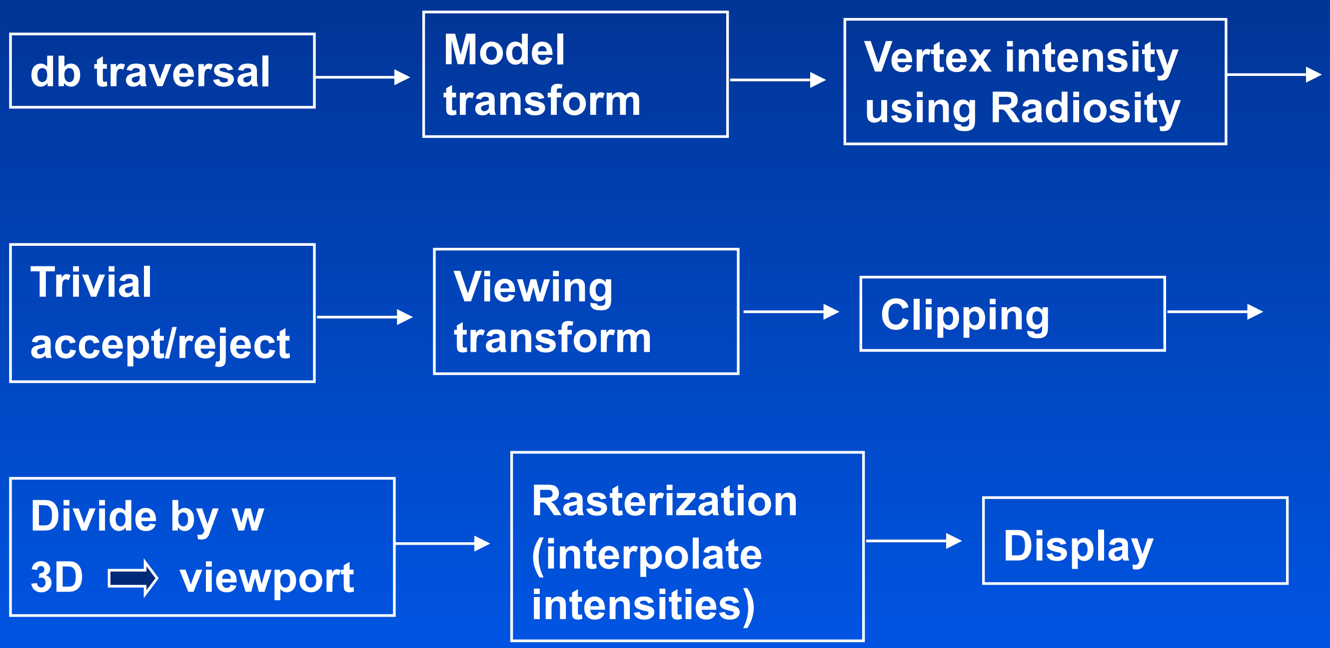


- Do trivial accept/reject before lighting to save time calculating illumination, e.g. back facing
- Do lighting before viewing x-form to preserve correct geometry

Z-buffer, Phong shading

- Lighting need to be done during rasterization (since normals are interpolated)
- To find R and V, need to inverse transform to world coordinate (not necessary if R and V does not change across a surface)

Radosity, Gouraud shading



Ray tracing

db traversal → Modeling transform → Ray tracing → Display

- Ray tracing are:
  - Hidden surface
  - Illumination
  - Rasterization
  - Perspective x-form♪