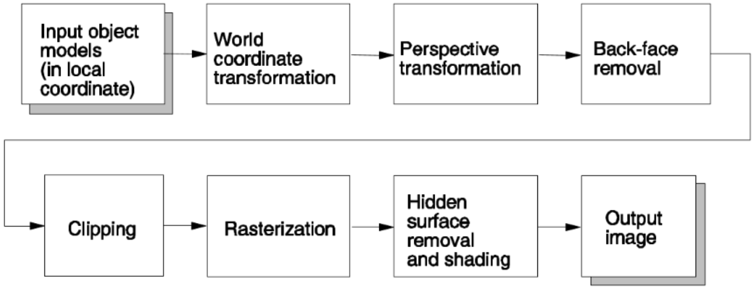
CS5182 Computer Graphics
Ray Tracing and Radiosity

2024/25 Semester A

City University of Hong Kong (DG)

Two Rendering Approaches

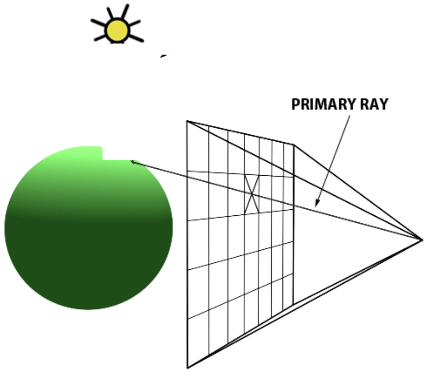
- Start from geometry
  - For each polygon/triangle:
    - Is it visible?
    - Where is it?
    - What color is it?


Input object models (in local coordinate) → World coordinate transformation → Perspective transformation → Back-face removal → Clipping → Rasterization → Hidden surface removal and shading → Output image

Rasterization rendering

  - It is extremely fast. However, it is hard to compute accurate shadows, reflections and refractions.
  - It cannot handle the scattering between objects.

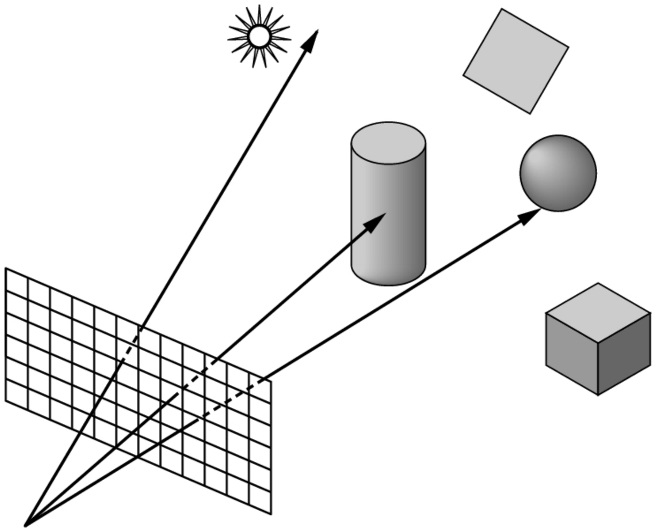
- Start from pixels
  - For each pixel in the final image
    - Which object is visible at this pixel?
    - What color is it?



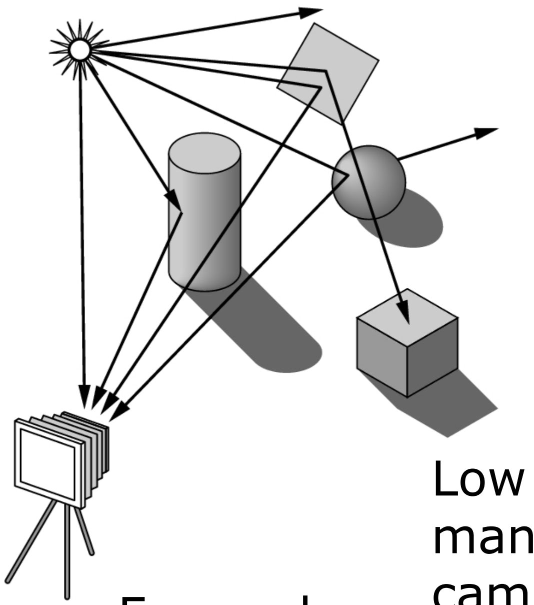
Ray tracing

Ray Tracing

- Ray-tracing generates an image by tracing the flow of rays in a scene, typically in a **backward** manner, from the eye point through the surfaces to the light sources.



Backward



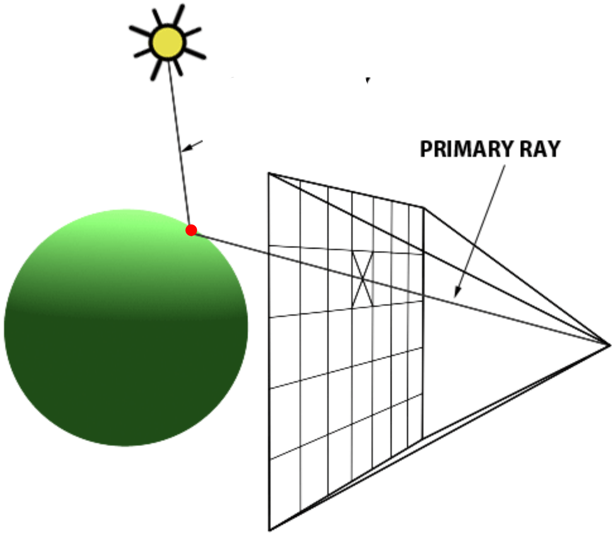
Forward

Low efficiency since many rays will miss camera and not contribute to image!

Ray Tracing

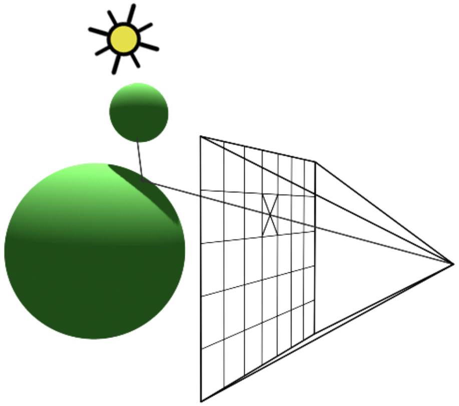
- For each pixel on the image plane, a ray is projected from the center of projection through the pixel into the scene. The first object that the ray intersects is determined.

- The point at which the ray hits the object (i.e., the ray intersection point) is also determined. The color value is then computed according to the surface normal and the light source information.

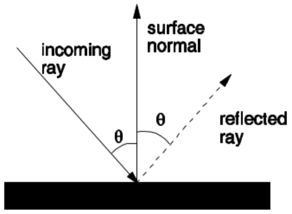


Ray Tracing

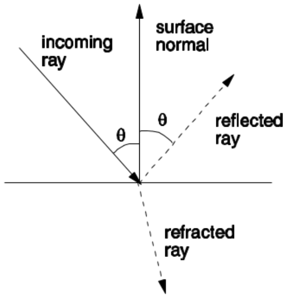
- In case if there is some object located between a particular light source and the ray intersection point, then this point is in shadow and the light contribution from that particular light source is not considered.



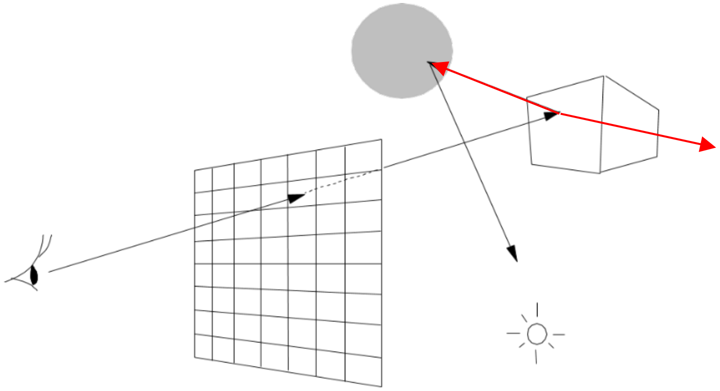
- The ray is then reflected and the process repeats. However, if the surface is a transparent surface, the ray is refracted as well as reflected.



Normal surface

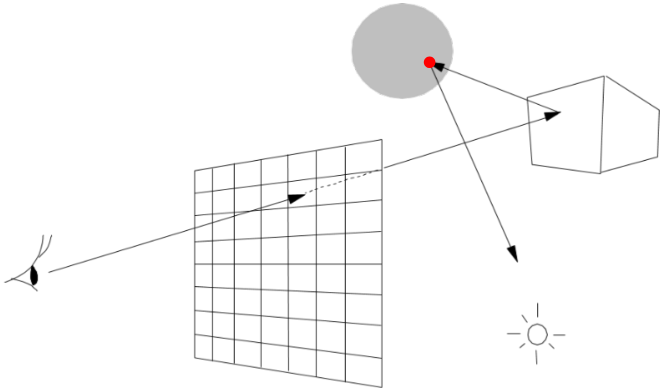


Transparent surface



Ray Tracing

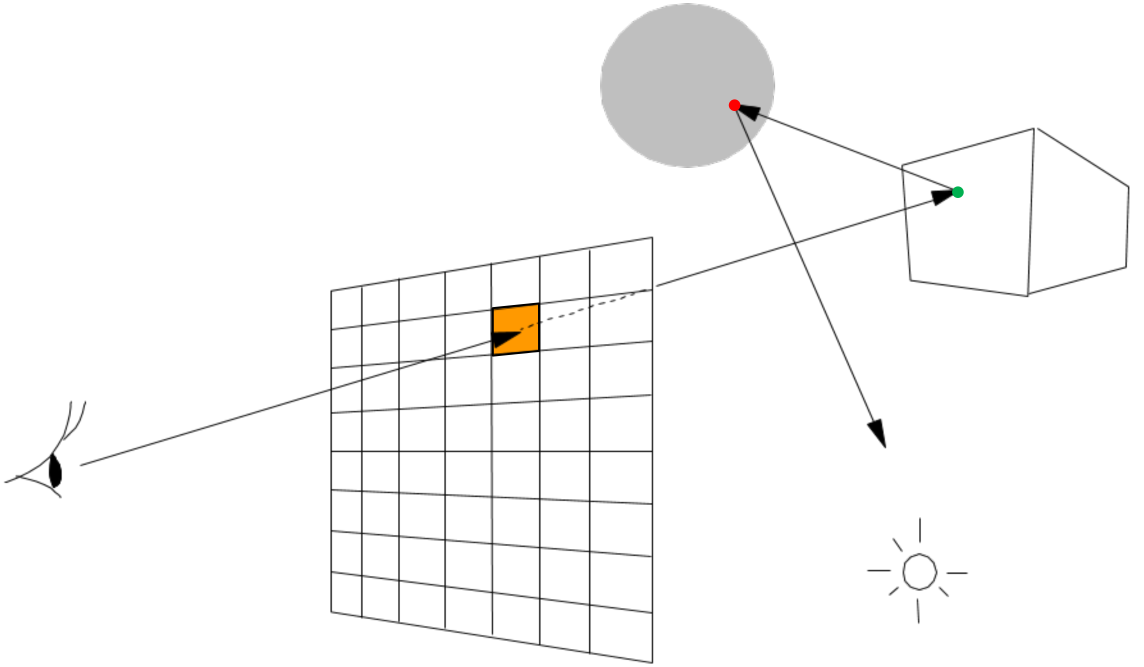
- The point at which the reflected/refracted ray hits the second object is again determined and the color value is also computed in a similar way as the first object.



- This process will continue until either we have reached a certain number of reflections or the color contribution made by the subsequent intersected objects has become too small. In practice, we may specify a maximum number of reflections that a ray can make to prevent spending too much processing time on a particular pixel.

Ray Tracing

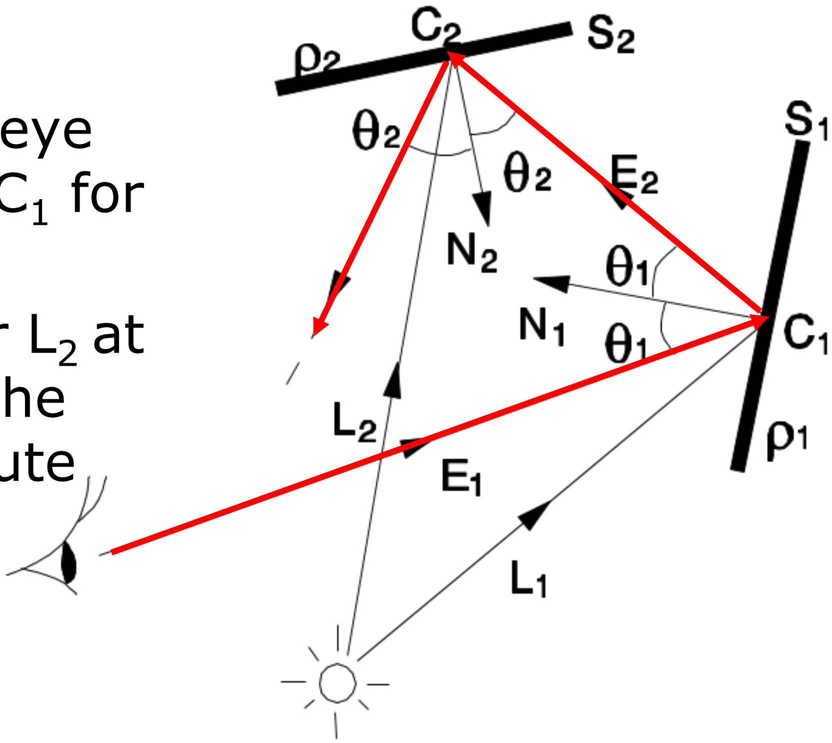
- The color values computed from all the intersected objects of a ray are weighted by the attenuation factors (which depend on the surface reflectivity) and added up to produce a single color value to become the color for the pixel.



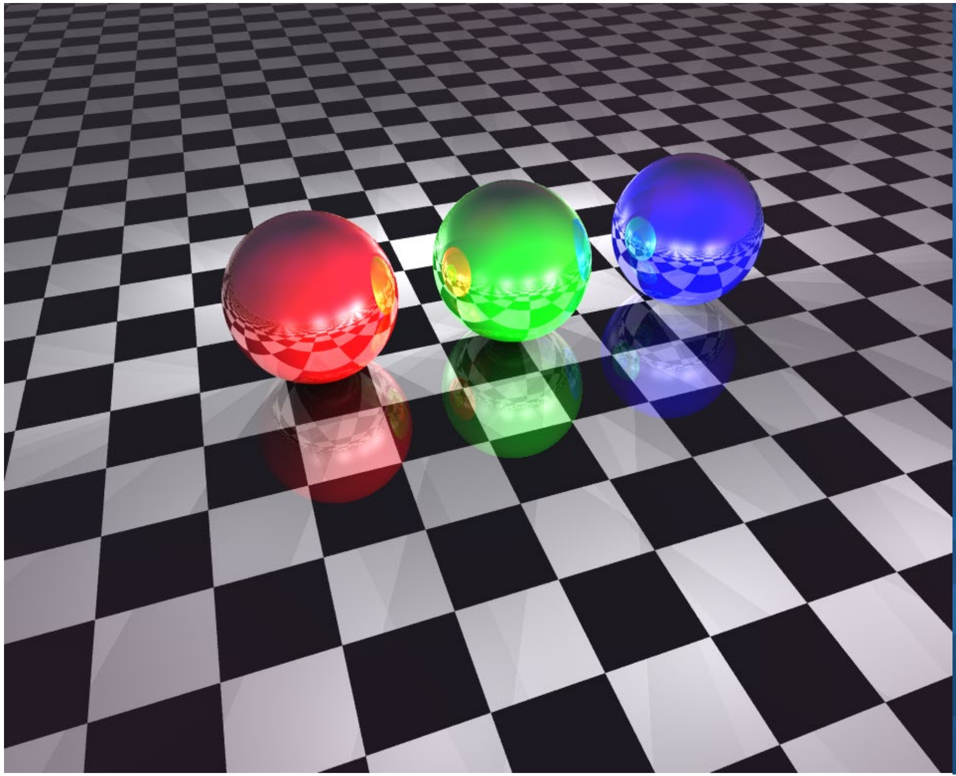
Ray Tracing

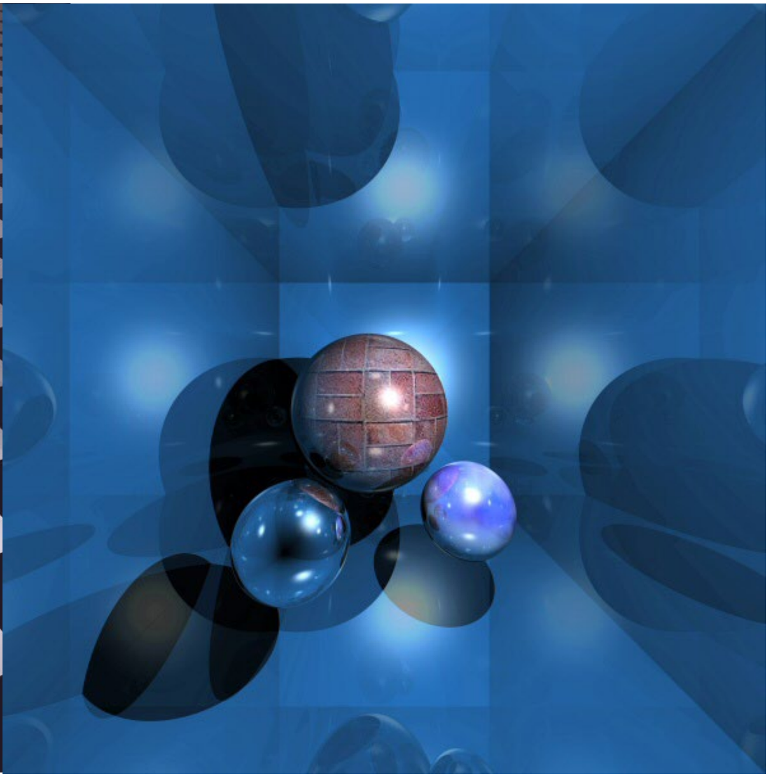
- Calculate the color of a pixel
  - Let the reflective indices of object primitives S_1 and S_2 be ⍰_1 and ⍰_2, respectively. Let also their surface normals be N_1 and N_2.
  - Given the light vector L_1 at the intersection point of S_1 by the eye ray E_1, we may compute color C_1 for the intersection point.
  - Similarly, given the light vector L_2 at the intersection point of S_2 by the reflected ray E_2, we may compute color C_2 for the point.
  - The color for the pixel is then computed as:

$$C_p = \text{⍰}_1 C_1 + \text{⍰}_1\text{⍰}_2 C_2 + \ldots$$



Ray Tracing





Ray Tracing

- Advantages
  - As ray-tracing follows the flow of rays through the scene, it may consider direct **specular** and direct diffuse reflections as well as indirect **specular** reflection.
  - It considers light refraction and shadowing.
- Disadvantages
  - It is slow due to the large number of costly ray-surface intersection tests and color calculations.
  - It does not consider indirect diffuse reflections, which involve diffuse reflections between surfaces. This will be discussed in ***Radiosity***.

Ray Tracing Acceleration

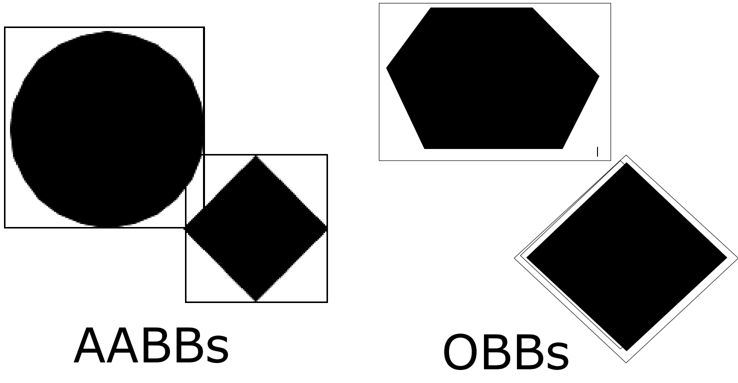
- Since ray-tracing spends a lot of time on detecting ray-object intersections, reducing the number of intersection tests can significantly reduce the computation time.
  - Bounding volumes
  - Space subdivision

Ray Tracing Acceleration

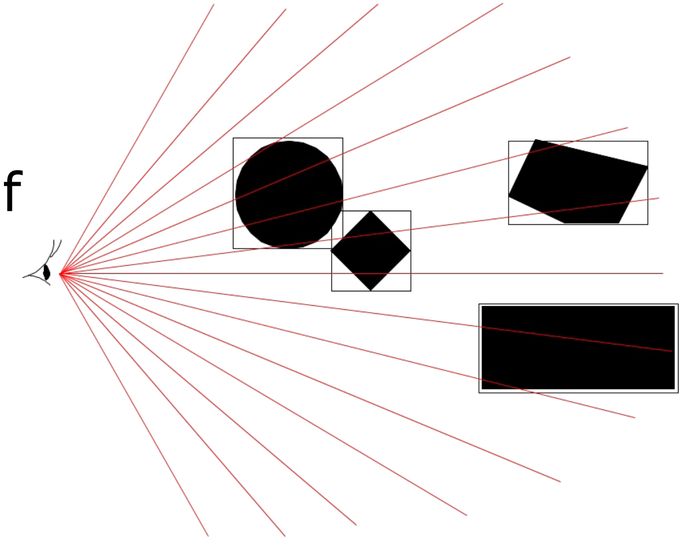
- Bounding volumes
  - In the bounding volume scheme, a bounding volume is constructed for each object in the scene.

  -- *axis-aligned bounding boxes (AABBs)*

  -- *oriented bounding boxes (OBBs)*

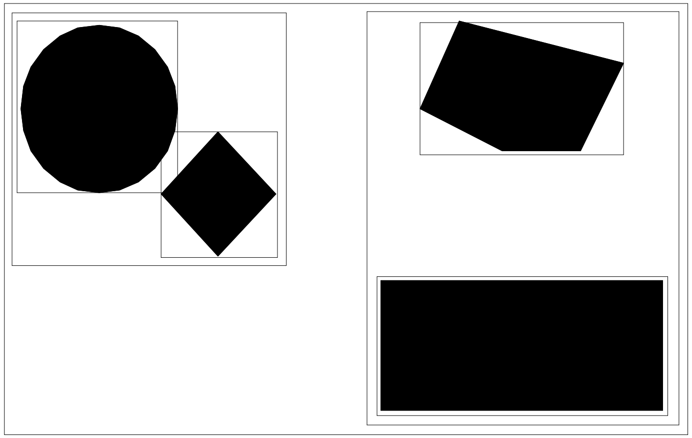


  - Instead of computing the intersection between each ray with each object, we first check if the ray intersects with the bounding volume of the object, which is much more efficient to do.

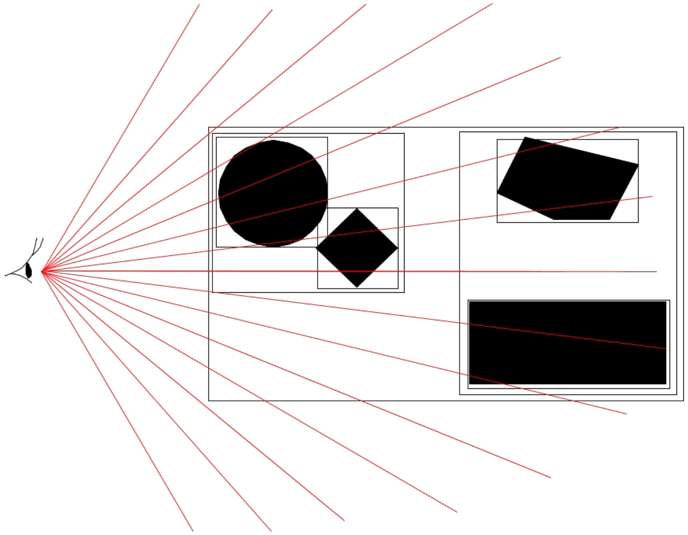


Ray Tracing Acceleration

- Bounding volumes
  - Further speedup can be achieved if we construct a hierarchical bounding volume of objects.



  - if a ray does not intersect with a bounding volume at whatever level, then the ray does not intersect with any objects bounded by its sub-volumes.



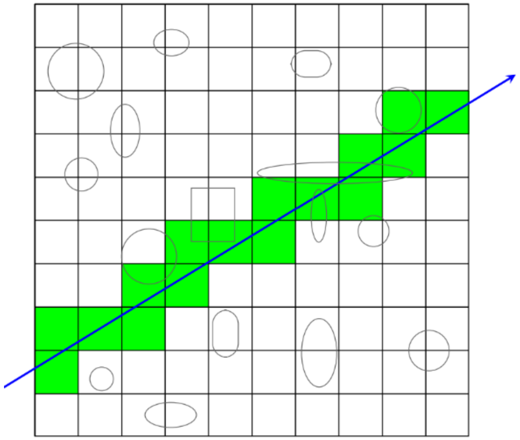
Ray Tracing Acceleration

- Space subdivision
  - Uniform space subdivision

    -- In each voxel, store the list of objects that intersects with the voxel

    -- Ray traverses the regular grid

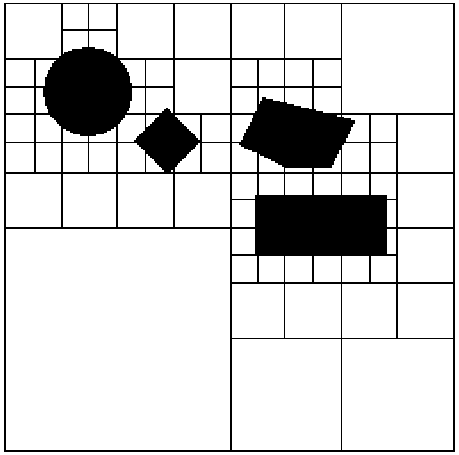
    -- For each traversed voxel, intersection test with stored objects



Uniform

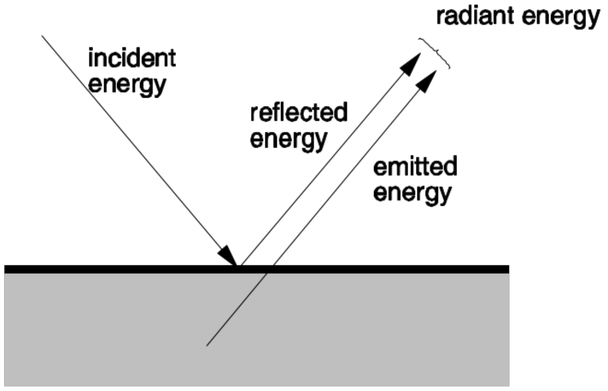
  - Non-uniform space subdivision

    --Subdivides the scene hierarchically (e.g., octree) to obtain fewer voxels.



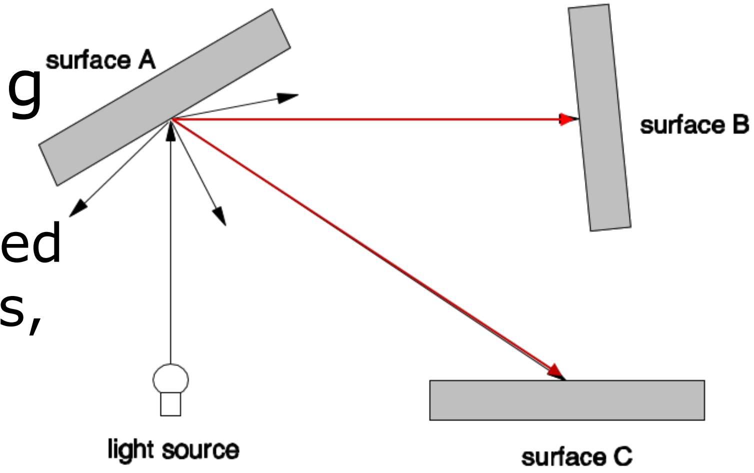
Non-uniform

Radiosity



- Radiosity is the radiant flux leaving (emitted and reflected by) a surface per unit area.

- The radiosity method is aimed at modeling **diffuse reflection** (both direct and indirect) among the surfaces in the scene.
  - In diffuse reflection, light is reflected with equal intensity in all directions, but the amount of light energy received by a surface depends on the orientation of the surface relative to the source.



-- **A** reflects light with equal intensity to **B** and **C**. **B** would receive higher light energy than **C** because **B** has a smaller incident angle.

Radiosity

- The radiosity B_i of a patch i is computed as:

$$B_i = E_i + \rho_i \sum_{j \in \text{all patches}} L_{i \leftarrow j}$$

where $L_{i \leftarrow j}$ is the amount of light from patch j to patch i, E_i is the light emitted from patch i, and ρ_i is the reflectivity of patch i.

- $L_{i \leftarrow j}$ can be further defined as follows:

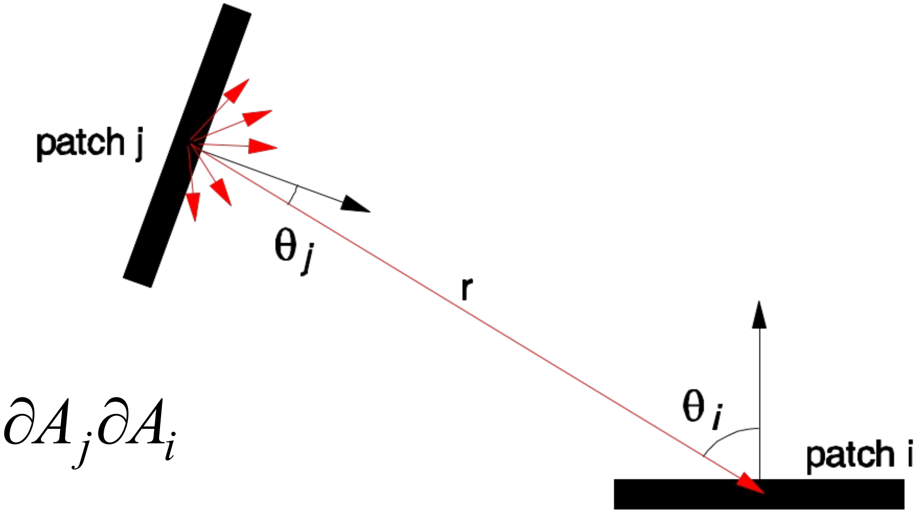
$$L_{i \leftarrow j} = F_{i \leftarrow j} B_j$$

where $F_{i \leftarrow j}$ is the form factor relating patch j to patch i, and

B_j is the radiosity of patch j.

Radiosity

- $F_{i \leftarrow j}$ is computed as

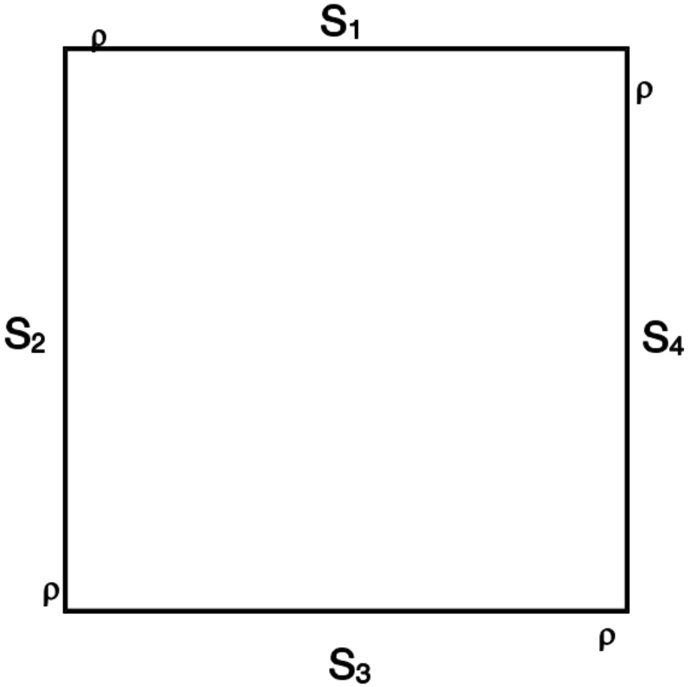


$$F_{i \leftarrow j} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} V_{ij} \partial A_j \partial A_i$$

where A_i and A_j are the areas of patches i and j. V_{ij} returns the visibility between patches i and j, i.e., V_{ij}=1 if there is no occlusion between them; V_{ij}=0 otherwise.

Radiosity

- Example
  - Consider an enclosed region with four polygons. For simplicity, we assume that all four polygons have the same reflective index ρ, where $0 \leq \rho \leq 1$. In fact, ρ is typically small for diffuse surfaces. S_1 is a light source. At time t, the light source is off. Hence, there is no light energy inside the enclosed region at this moment.



Radiosity

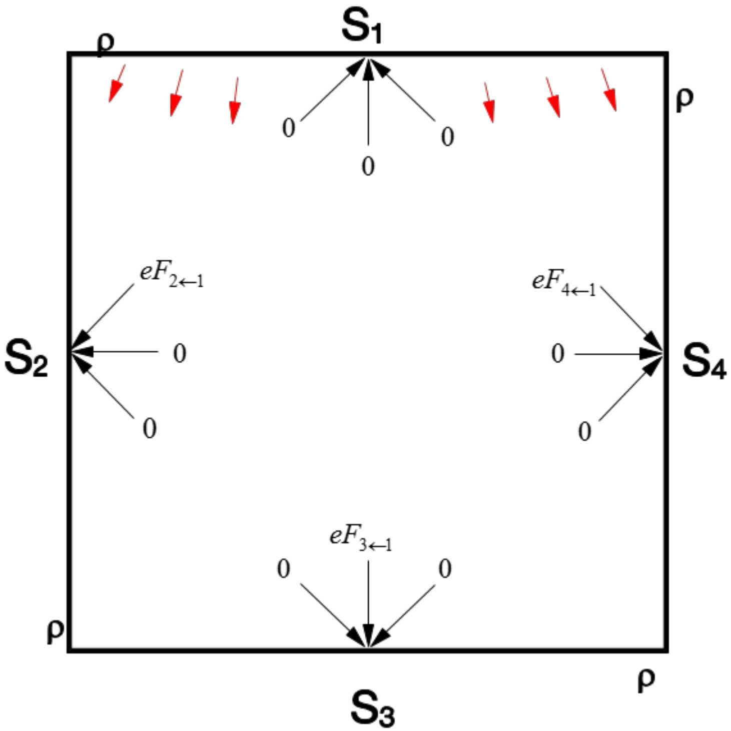
- Example
  - At time $t+1$, we turn on S_1 with an energy e. the total energy received by each of the polygons is as follows:

$$S_1:\ L_1 = 0$$

$$S_2: L_2 = eF_{2\leftarrow 1}$$

$$S_3: L_3 = eF_{3\leftarrow 1}$$

$$S_4: L_4 = eF_{4\leftarrow 1}$$



Radiosity

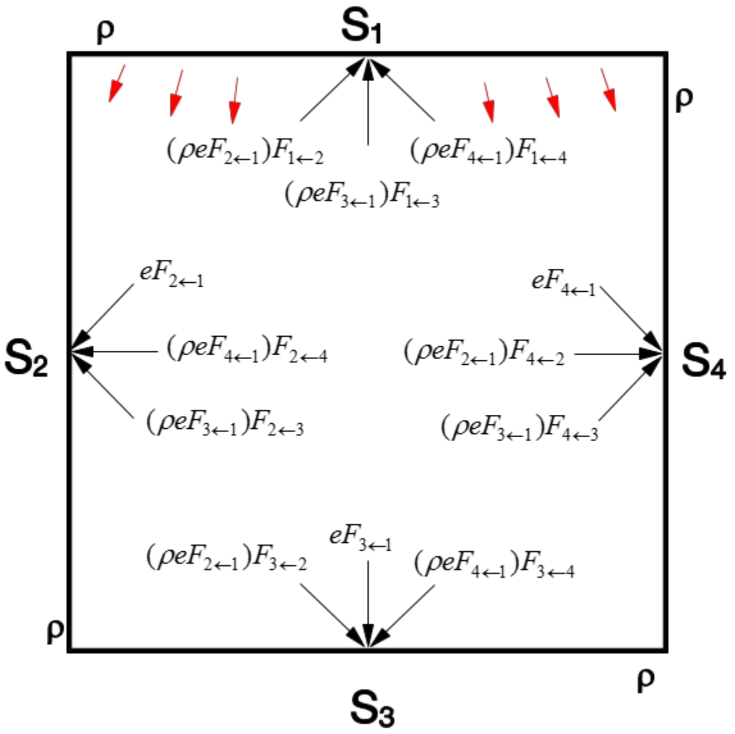
- Example
  - At time $t+2$, the total energy received by each of the polygons is as follows:

$$\mathrm{S}_1: \quad L_1 = \rho e(F_{2\leftarrow1}F_{1\leftarrow2} + F_{3\leftarrow1}F_{1\leftarrow3} + F_{4\leftarrow1}F_{1\leftarrow4})$$

$$\mathrm{S}_2: \quad L_2 = eF_{2\leftarrow1} + \rho eF_{4\leftarrow1}F_{2\leftarrow4} + \rho eF_{3\leftarrow1}F_{2\leftarrow3}$$

$$\mathrm{S}_3: \quad L_3 = eF_{3\leftarrow1} + \rho eF_{2\leftarrow1}F_{3\leftarrow2} + \rho eF_{4\leftarrow1}F_{3\leftarrow4}$$

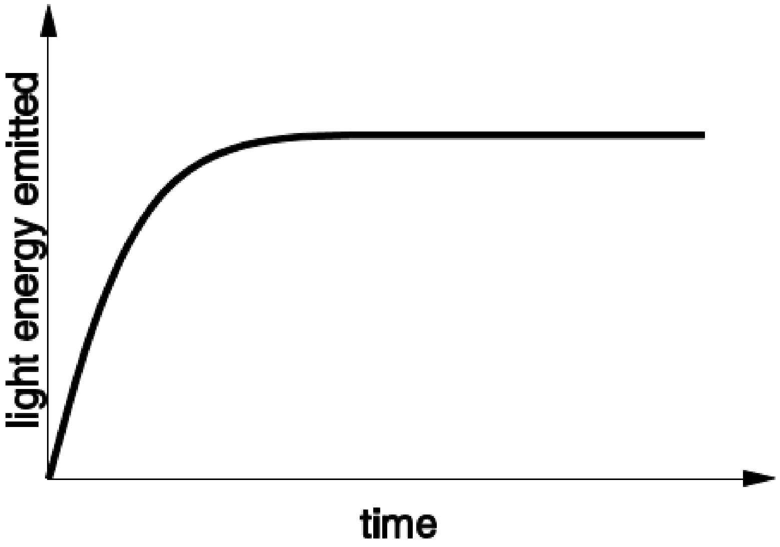
$$\mathrm{S}_4: \quad L_4 = eF_{4\leftarrow1} + \rho eF_{2\leftarrow1}F_{4\leftarrow2} + \rho eF_{3\leftarrow1}F_{4\leftarrow3}$$



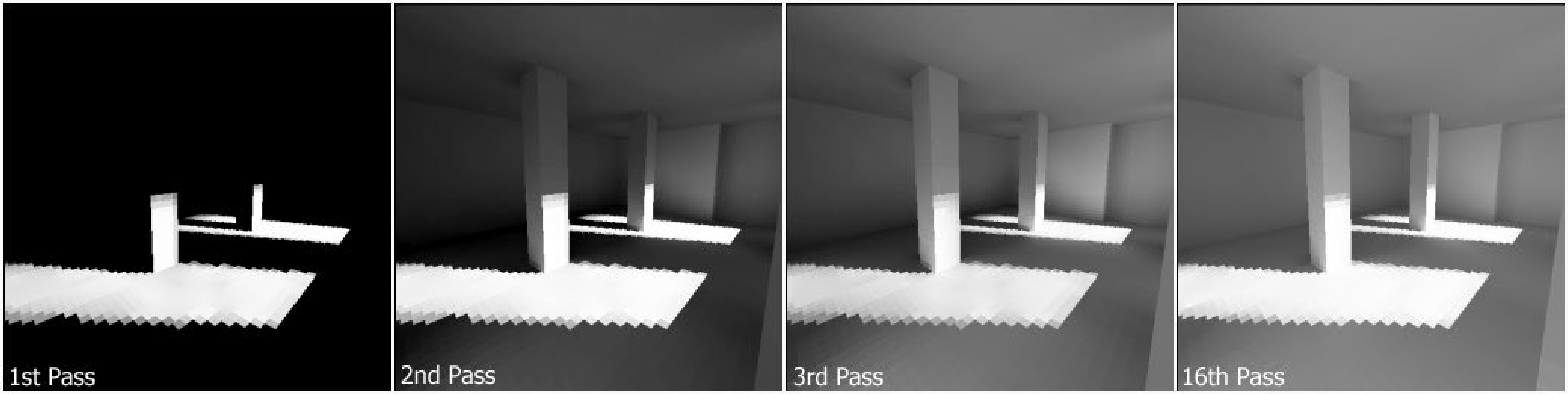
ρ
S₁
ρ
$(\rho eF_{2\leftarrow1})F_{1\leftarrow2}$
$(\rho eF_{4\leftarrow1})F_{1\leftarrow4}$
$(\rho eF_{3\leftarrow1})F_{1\leftarrow3}$
$eF_{2\leftarrow1}$
$eF_{4\leftarrow1}$
S₂
$(\rho eF_{4\leftarrow1})F_{2\leftarrow4}$
$(\rho eF_{2\leftarrow1})F_{4\leftarrow2}$
S₄
$(\rho eF_{3\leftarrow1})F_{2\leftarrow3}$
$(\rho eF_{3\leftarrow1})F_{4\leftarrow3}$
$eF_{3\leftarrow1}$
$(\rho eF_{2\leftarrow1})F_{3\leftarrow2}$
$(\rho eF_{4\leftarrow1})F_{3\leftarrow4}$
ρ
ρ
S₃

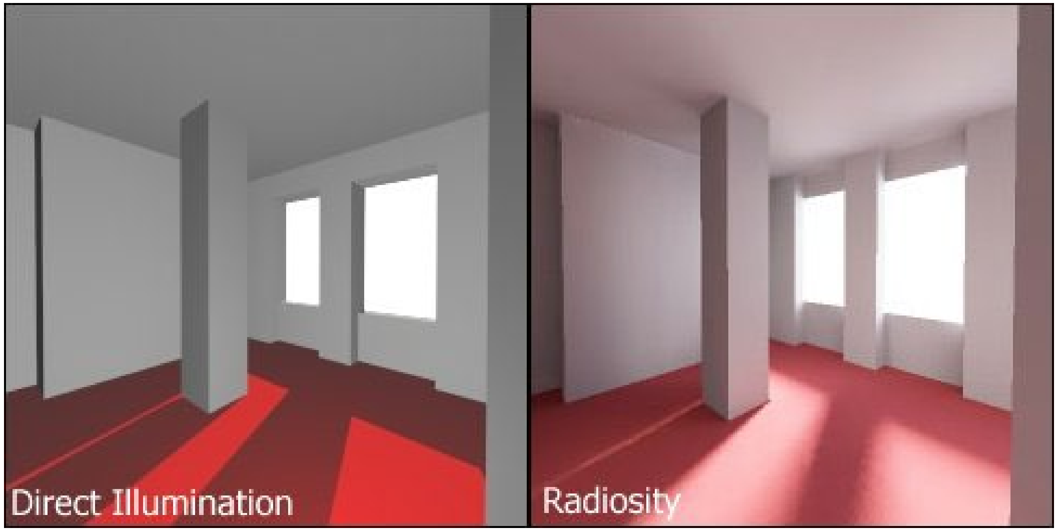
Radiosity

- Example
  - There should be more and more terms in each equation as we increase t.
  - However, as ρ is normally smaller than 1, the subsequent terms will become smaller and smaller due to the increasing number of reflections.
  - Hence, the emitted energy, i.e., radiosity, from each surface should converge to a certain value.



Radiosity


1st Pass
2nd Pass
3rd Pass
16th Pass


Direct Illumination
Radiosity

Radiosity

- Radiosity for a large surface
  - For a large surface, the radiosity value may change significantly from one end to the other, due to the change in the angle of incidence.
  - To improve the accuracy, surfaces in the scene are divided into small patches. Each patch is assumed to have a finite size, emitting and reflecting light uniformly over its entire patch area.

Radiosity

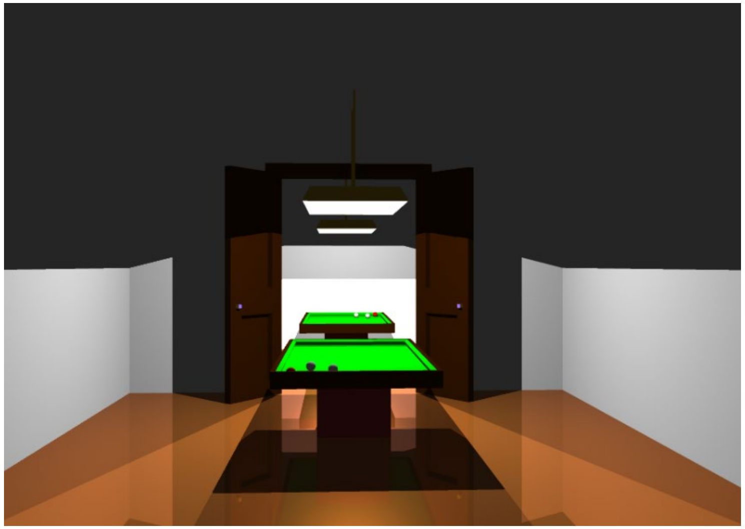
- Radiosity for a large surface
  - Assuming that there are a total of n patches in the scene. We now have n simultaneous equations with n unknowns (B_i). We may construct a matrix to solve them.

$$\begin{bmatrix} B'_1 \\ B'_2 \\ \vdots \\ B'_i \\ \vdots \\ B'_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_i \\ \vdots \\ E_n \end{bmatrix} + \begin{bmatrix} \rho_1 F_{1\leftarrow 1} & \rho_1 F_{1\leftarrow 2} & \cdots & \rho_1 F_{1\leftarrow j} & \cdots & \rho_1 F_{1\leftarrow n} \\ \rho_2 F_{2\leftarrow 1} & \rho_2 F_{2\leftarrow 2} & \cdots & \rho_2 F_{2\leftarrow j} & \cdots & \rho_2 F_{2\leftarrow n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \rho_i F_{i\leftarrow 1} & \rho_i F_{i\leftarrow 2} & \cdots & \rho_i F_{i\leftarrow j} & \cdots & \rho_i F_{i\leftarrow n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \rho_n F_{n\leftarrow 1} & \rho_n F_{n\leftarrow 2} & \cdots & \rho_n F_{n\leftarrow j} & \cdots & \rho_n F_{n\leftarrow n} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix}$$

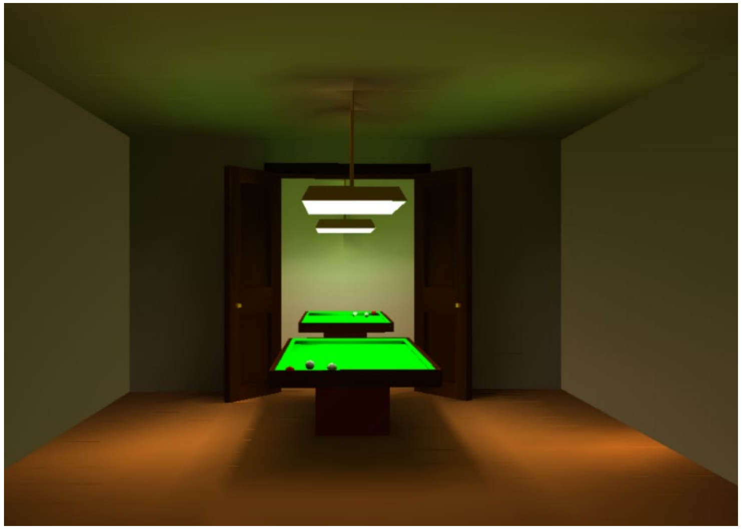
Radiosity

- Radiosity for a large surface
  - Once we have computed the radiosity values for all patches, we may use them to update the color values of the primitives. These primitives can then be rendered using a scan-conversion method.
  - Note that the computed radiosities are view-independent. Hence, the radiosity method cannot handle **specular reflection**, which is view-dependent.
  - To handle both, we may combine the radiosity method with the ray-tracing method. The price is a further increase in computation time.

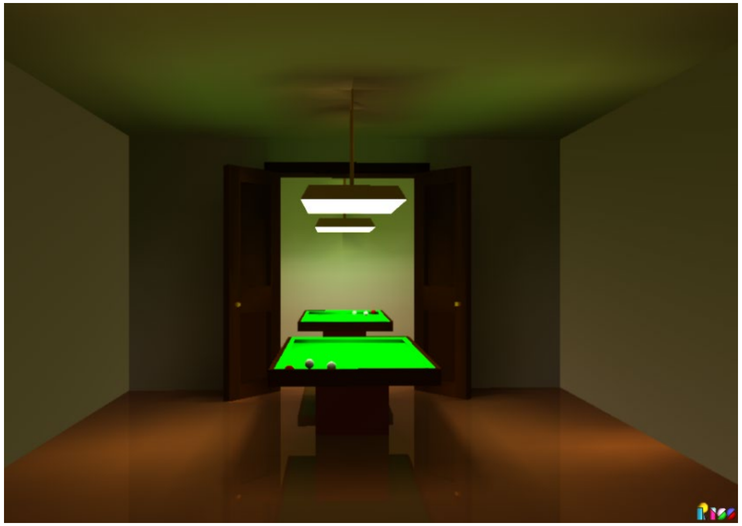
Comparison



Ray tracing



Radiosity



Ray tracing +Radiosity