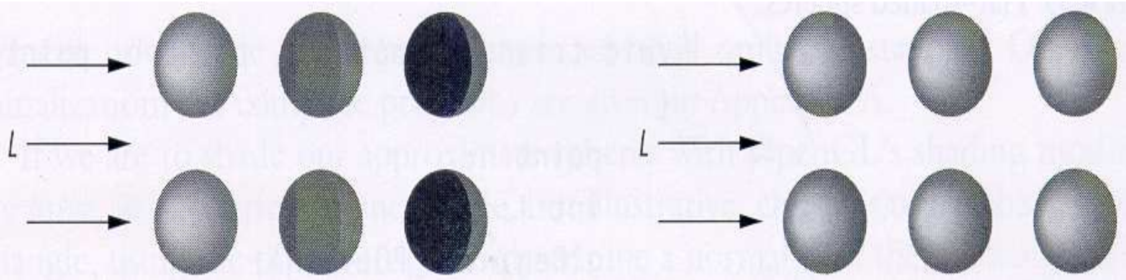
Global Shading Models



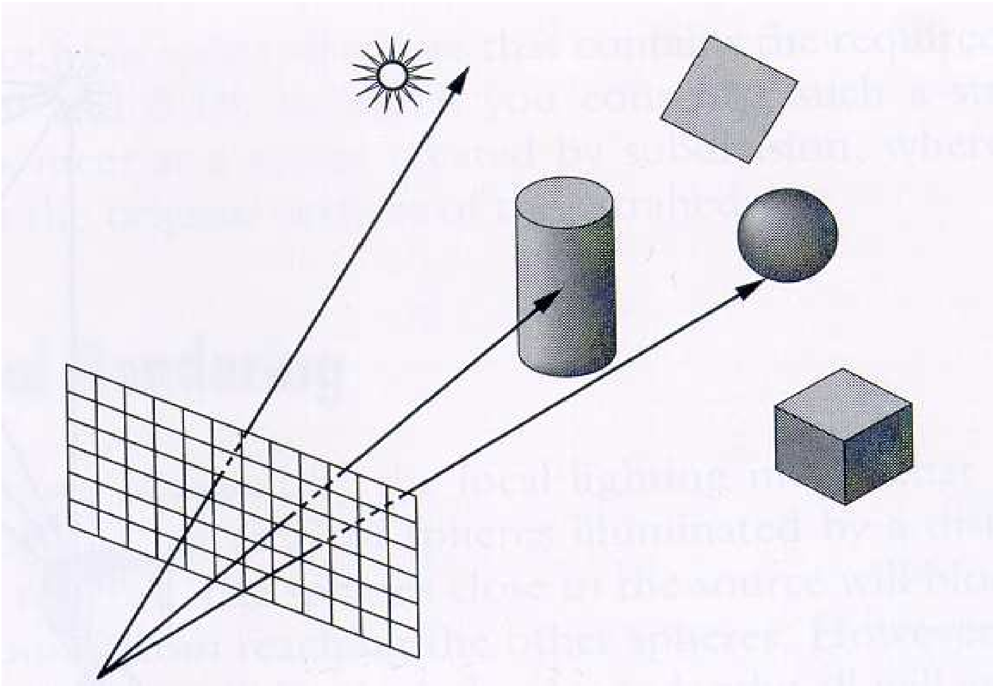
- **Ray tracing**

  - It is more efficient for specular and translucent surfaces

  - Shadows can also be produced

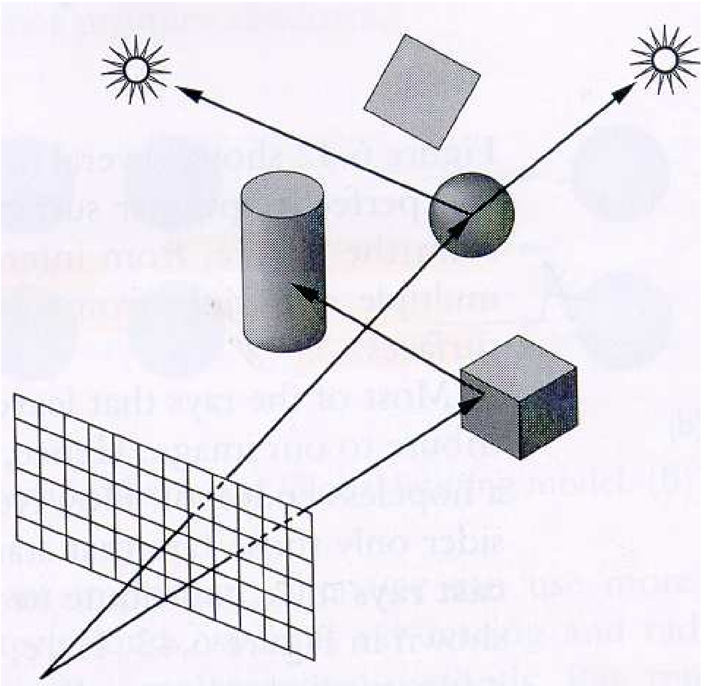
- **Steps involved in tracing**

  - Cast a ray through each pixel



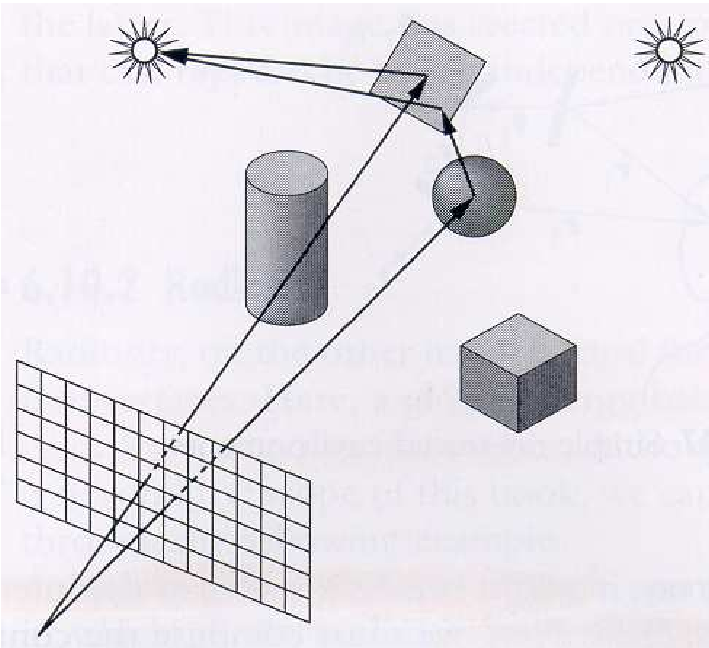
- For each intersection point, check whether it is illuminated

- If the ray from a point on a surface to the light source intersects another surface before it meets the source, then this point is in **shadow**

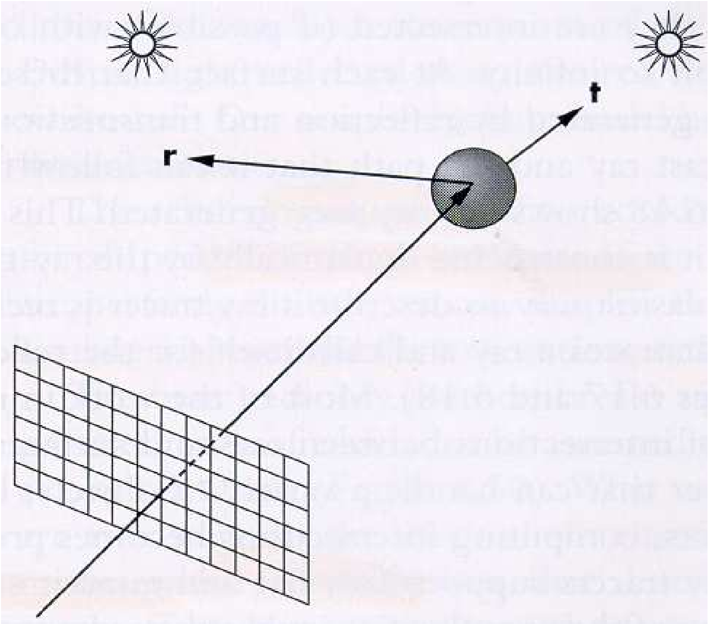


- If a surface is reflective, we can follow the ray as it bounces from surface to surface until it either goes off to infinity or intersects a source

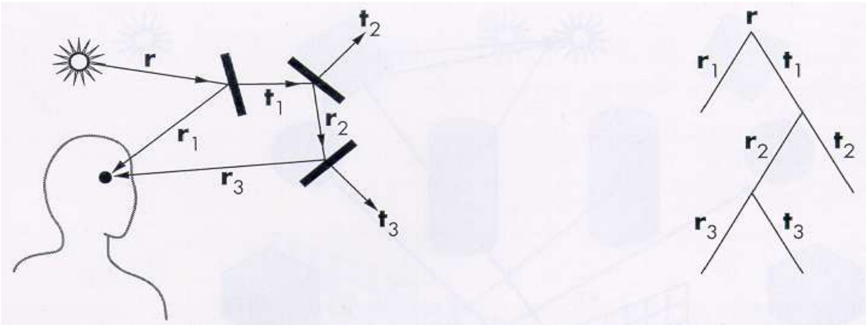
- Such calculations are done **recursively**, and take into account any absorption of light at surfaces



- If a surface is translucent, a ray is split into two rays (transmitted and reflected)



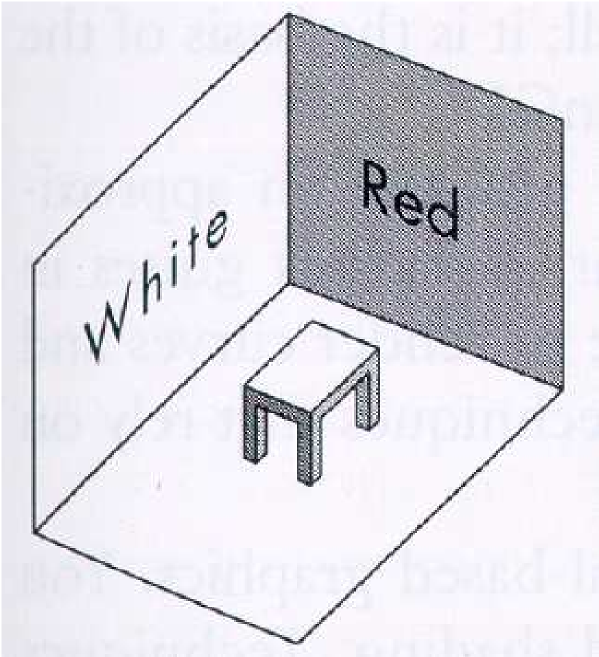
- Ray tree



- **Radiosity**

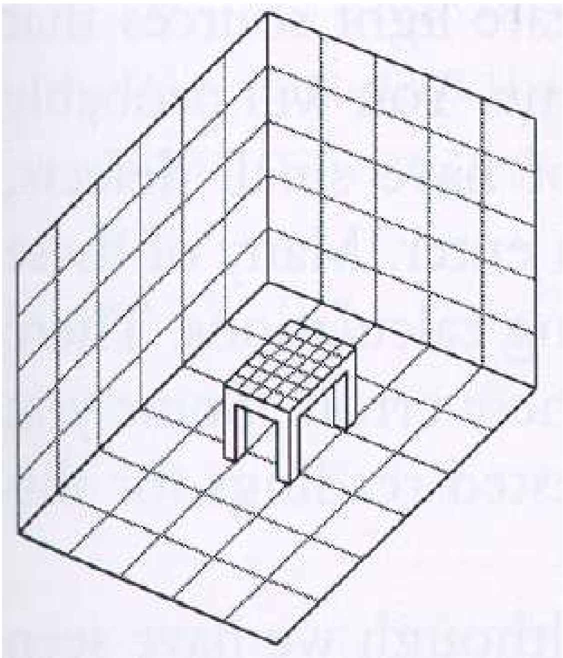
  - It is ideal for diffuse surfaces

  - Take into consideration reflections from other -diffuse- surfaces



  - Divide the scene into small flat polygons (patches)

  - Each patch will be shaded using constant shade



- Step 1

  * Consider patches pairwise to determine *form factors* (how the light energy leaving one patch affects the other)

  * This step takes $O(n^2)$ time (n: number of patches)

- Step 2

  * The *rendering equation* (integral equation) is reduced to a set of linear equations

  * Each equation involves the radiosities (reflectivity) of the patches

- Step 3

  * Render the scene using a local model

  * Results are independent from the position of the viewer