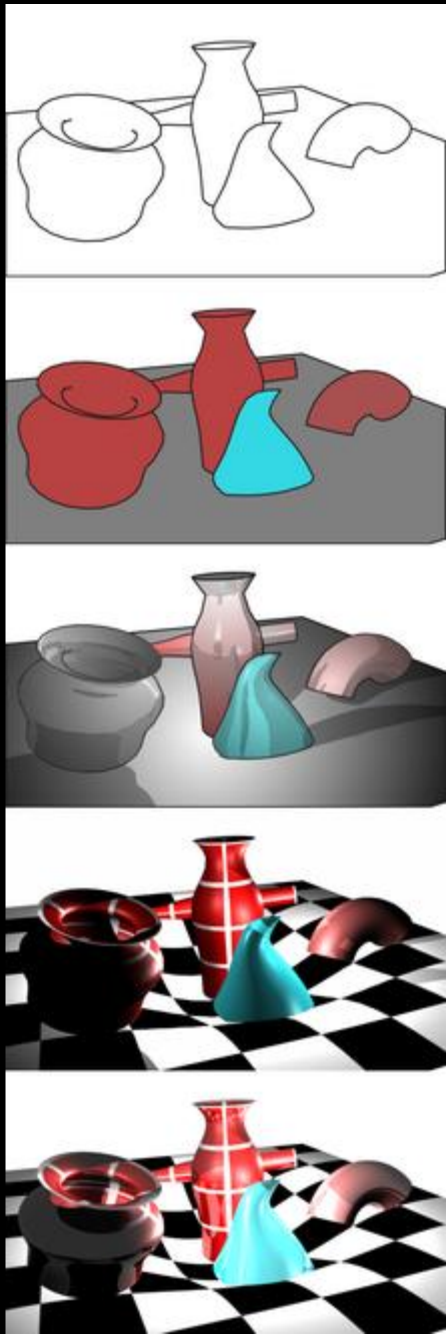


Rendering

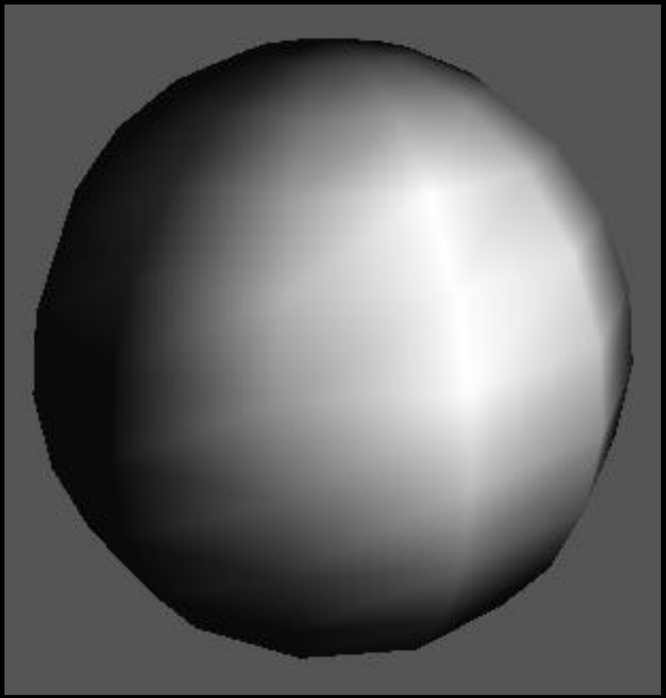
Rendering

- Rendering is the process of generating a photorealistic or non-photorealistic image from a 2D or 3D model by means of a computer program.

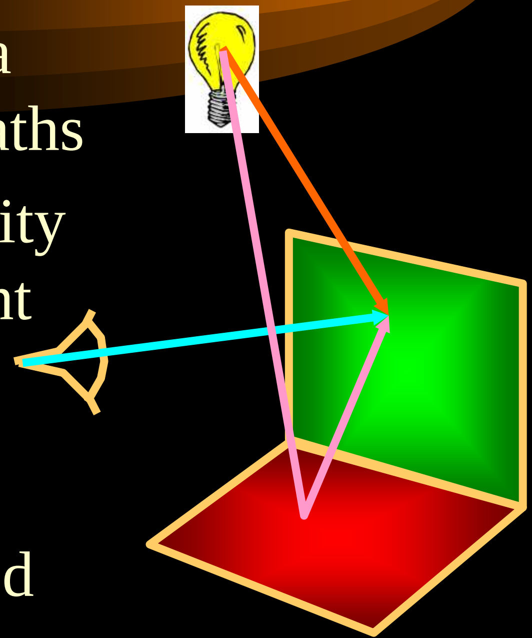
- Realistic display of a scene are obtained by generating perspective projections of objects and by applying natural lighting effects to the visible surfaces.

"Lighting"

- Two components:
  - **Lighting Model** or **Shading Model** - how to calculate the intensity at a point on the surface
  - **Surface Rendering Method** - How to calculate the intensity at each pixel

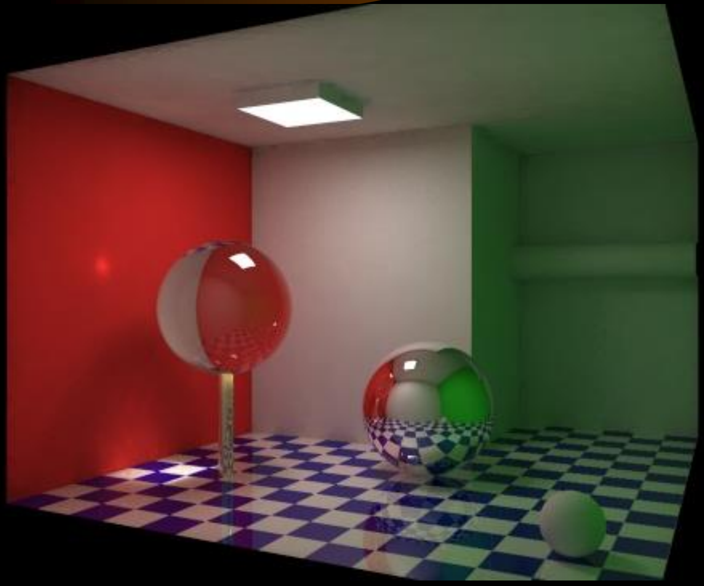


- Illumination - the transport of light from a source to a point via direct and indirect paths
- Lighting - computing the luminous intensity for a specified 3D point, given a viewpoint
- Shading - assigning colors to pixels
- Illumination Models:
  - Empirical - approximations to observed light properties
  - Physically based - applying physics properties of light and its interactions with matter



Lighting

- **Global illumination** – the transport of light within a scene.



Light Source – Source of light that enable to see the object

- Light emitting source type :

  - point source (sun, bulb,..)

  - Distributed source (tube light)

- Reflection source type:

  - Diffusion (reflect evenly in all direction; rough & grainy surface)

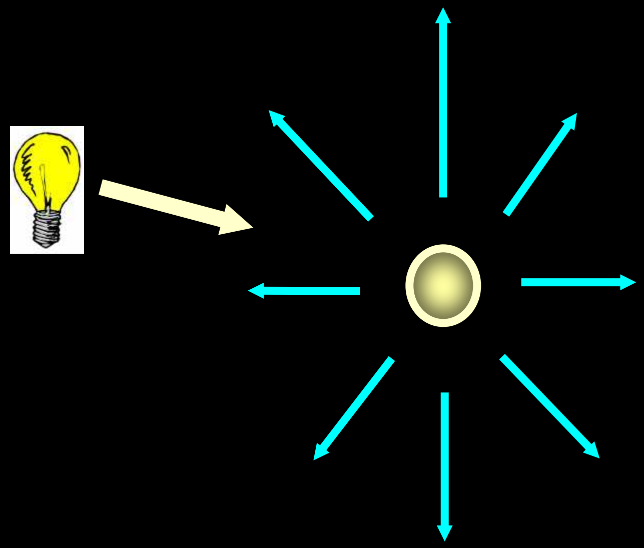
  - Specular (reflected from a spot or specific point; shiny, bright spot)

Two components

- Light Source Properties
  - Color (Wavelength(s) of light)
  - Shape
  - Direction
- Object Properties
  - Material
  - Geometry
  - Absorption

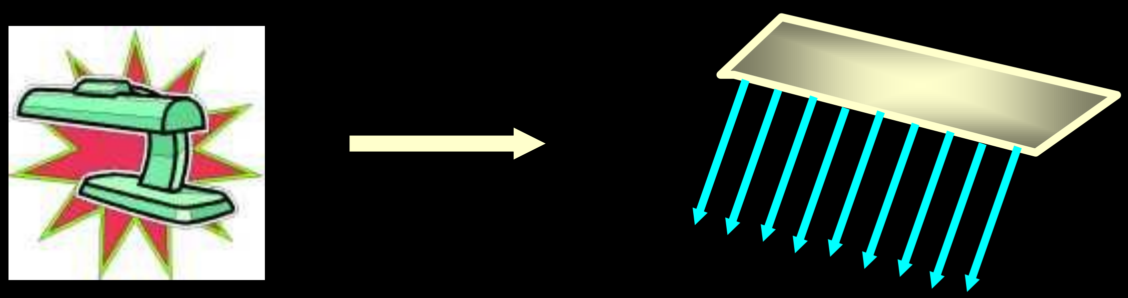
Light Source Properties

- Color
  - the light has one wavelength
- Shape
  - point light source - approximate the light source as a 3D point in space. Light rays emanate in all directions.
    - good for small light sources (compared to the scene)
    - far away light sources



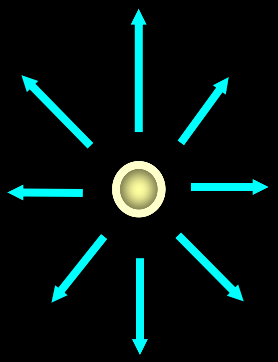
Distributed Lights

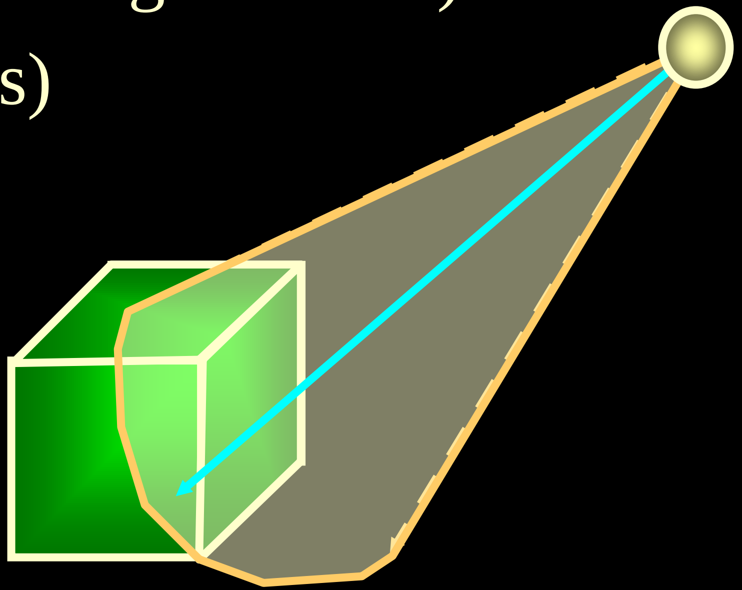
- Light Source Shape ….



Light Source Direction

- Usually lights are treated as *rays* emanating from a source. The *direction* of these rays can either be:
  - Omni-directional (point light source)
  - Directional (spotlights)





Light Position

- Specify the position of a light with *x*, *y*, and *z* coordinate.
  - These lights are called ***positional lights***
- Q: Are there types of lights that we can simplify?

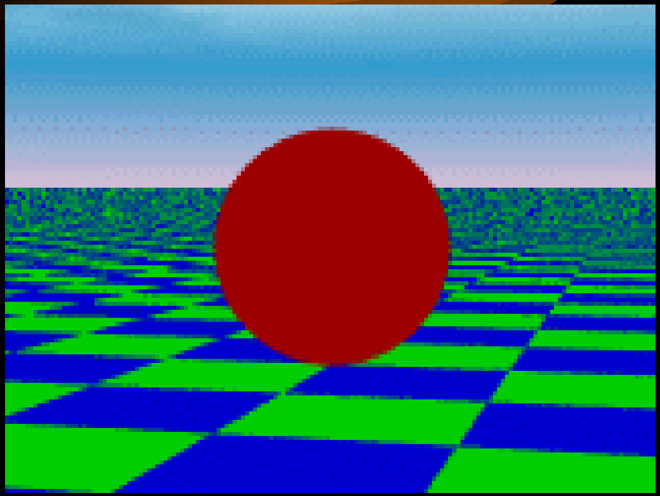
A: Consider sun. If a light is significantly far away, we can represent the light with *only* a direction vector. These are called ***directional lights.***

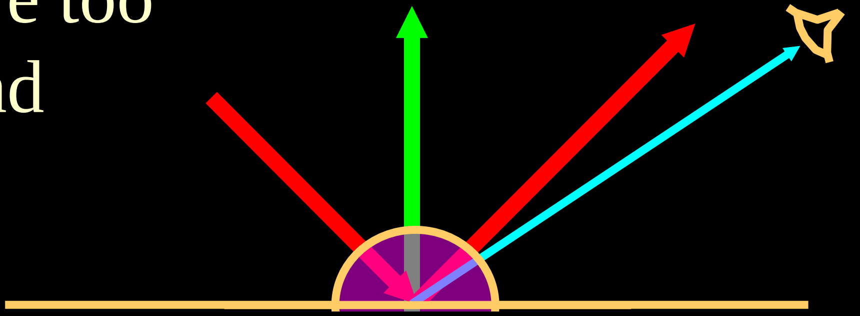
Contributions from lights

- Breakdown what a light does to an object into three different components. This APPROXIMATES what a light does. To actually compute the rays is too expensive to do in real-time.
  - Light at a pixel from a light = Ambient + Diffuse + Specular contributions.
  - $I_{light} = I_{ambient} + I_{diffuse} + I_{specular}$

Ambient Term - Background Light

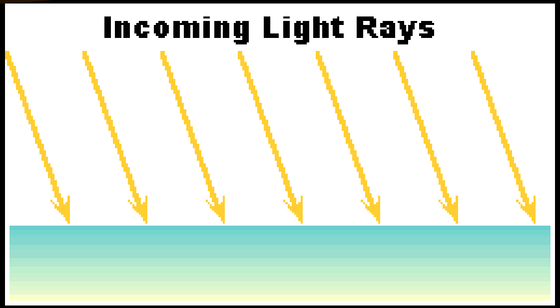
- It represents the approximate contribution of the light to the general scene, regardless of location of light and object
- Indirect reflections that are too complex to completely and accurately compute
- $I_{ambient}$ = color

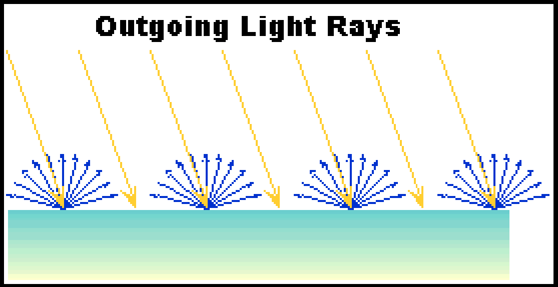




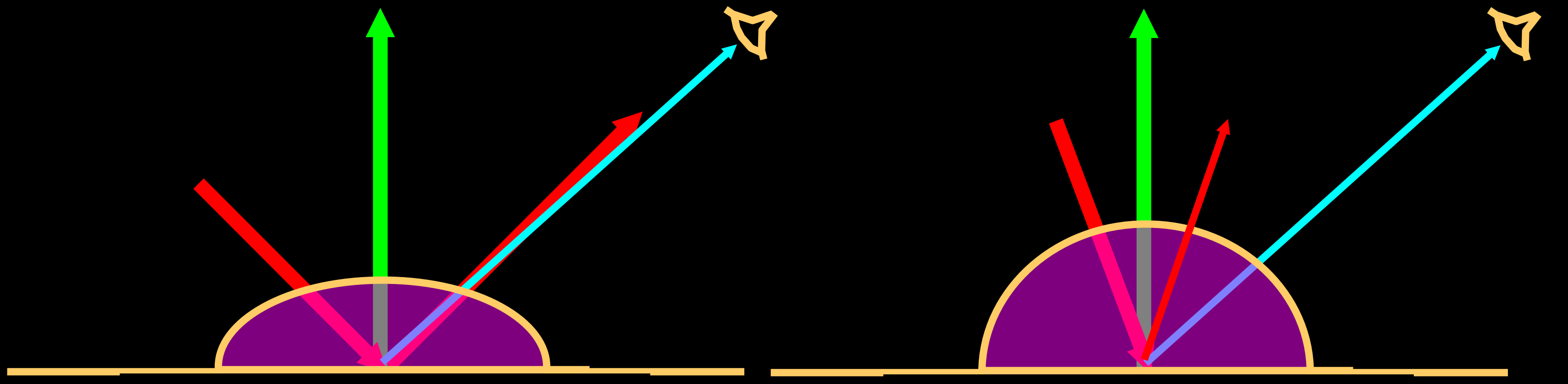
Diffuse Term

- Contribution that a light has on the surface, ***regardless of viewing direction***.
- Diffuse surfaces, on a microscopic level, are very rough. This means that a ray of light coming in has an equal chance of being reflected in *any* direction.
- Reflectivity – ratio of light reflected from the surface to the total incoming light. (White surface -1 ; black – 0)





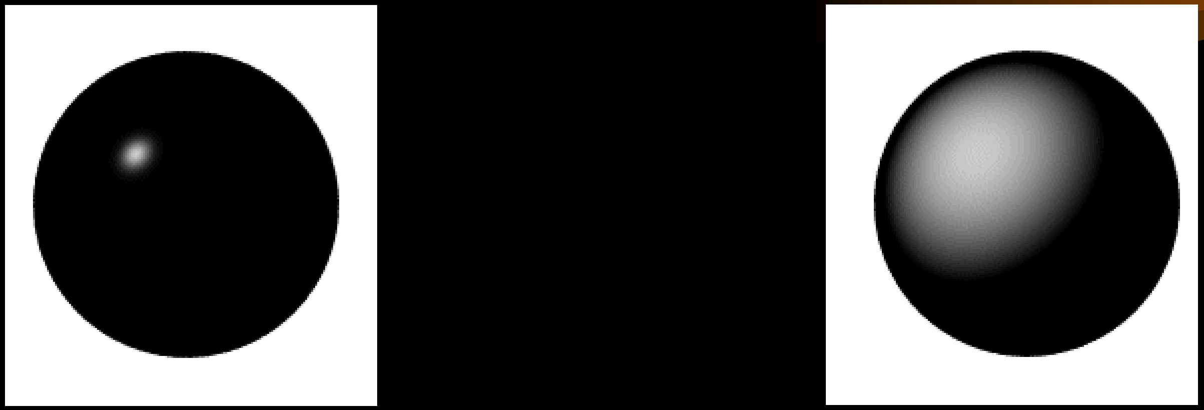
Example

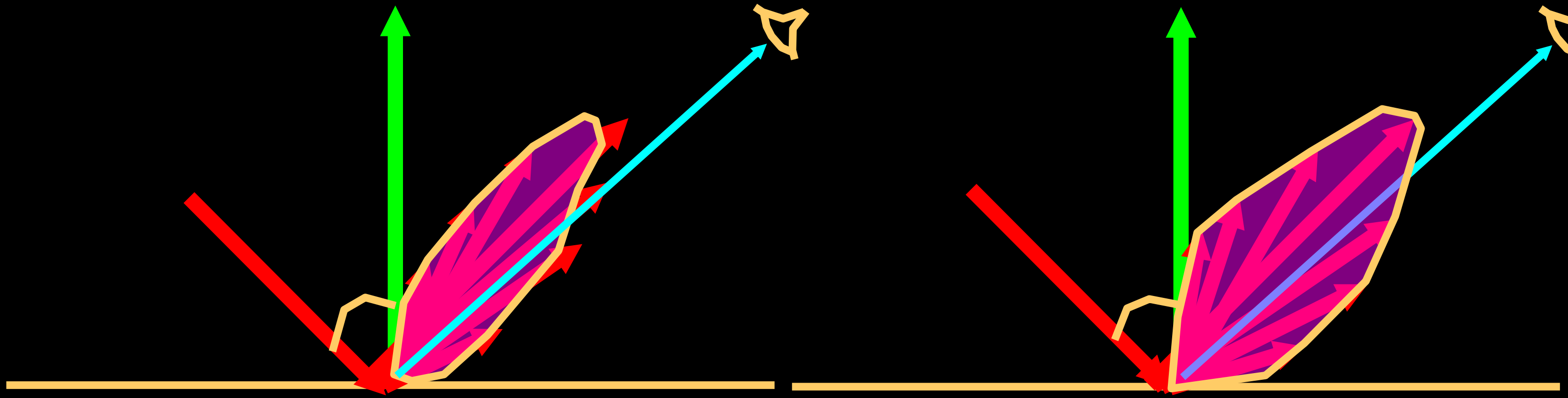


Specular Reflection

- Specular contribution can be thought of as the “shiny highlight” of a plastic object.
- On a microscopic level, the surface is very smooth. Almost all light is reflected.
- Reflection in single direction
- Angle of reflection: angle that the reflected beam makes with the surface normal
- Surface normal – objects (line, ray, vector) perpendicular to the surface.

Shiny vs. dull objects

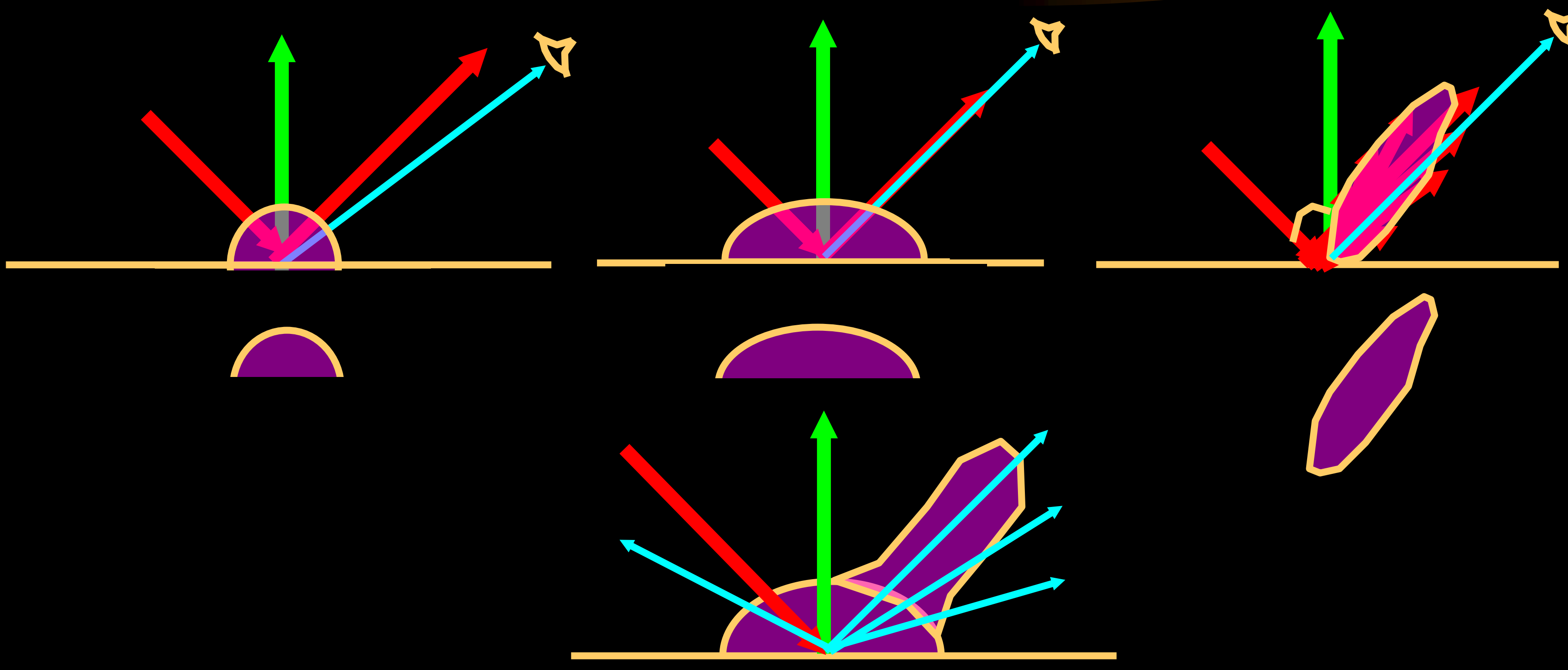




Combining the terms

- Ambient - the combination of light reflections from various surfaces to produce a uniform illumination. Background light.
- Diffuse - uniform light scattering of light rays on a surface. Proportional to the “amount of light” that hits the surface. Depends on the surface normal and light vector.
- Sepecular - light that gets reflected. Depends on the light ray, the viewing angle, and the surface normal.

Ambient + Diffuse + Specular



Full Illumination Model

I = i * K

Intensity = Light source * Reflectivity

$$I_{final} = I_{l_{ambient}} k_{ambient} + \sum_{l=0}^{lights-1} f(d_l)\left[I_{l_{diffuse}} k_{diffuse} (N \cdot L) + I_{l_{specular}} k_{specular} (N \cdot H)^{shininess} \right]$$

$$f(d) = \min\left(1, \frac{1}{a_0 + a_1 d + a_2 d^2}\right)$$

Shading

- Shading is how we “color”
- Shade any surface by calculating the surface normal at each visible point and applying the desired illumination pattern at that point.

Types:

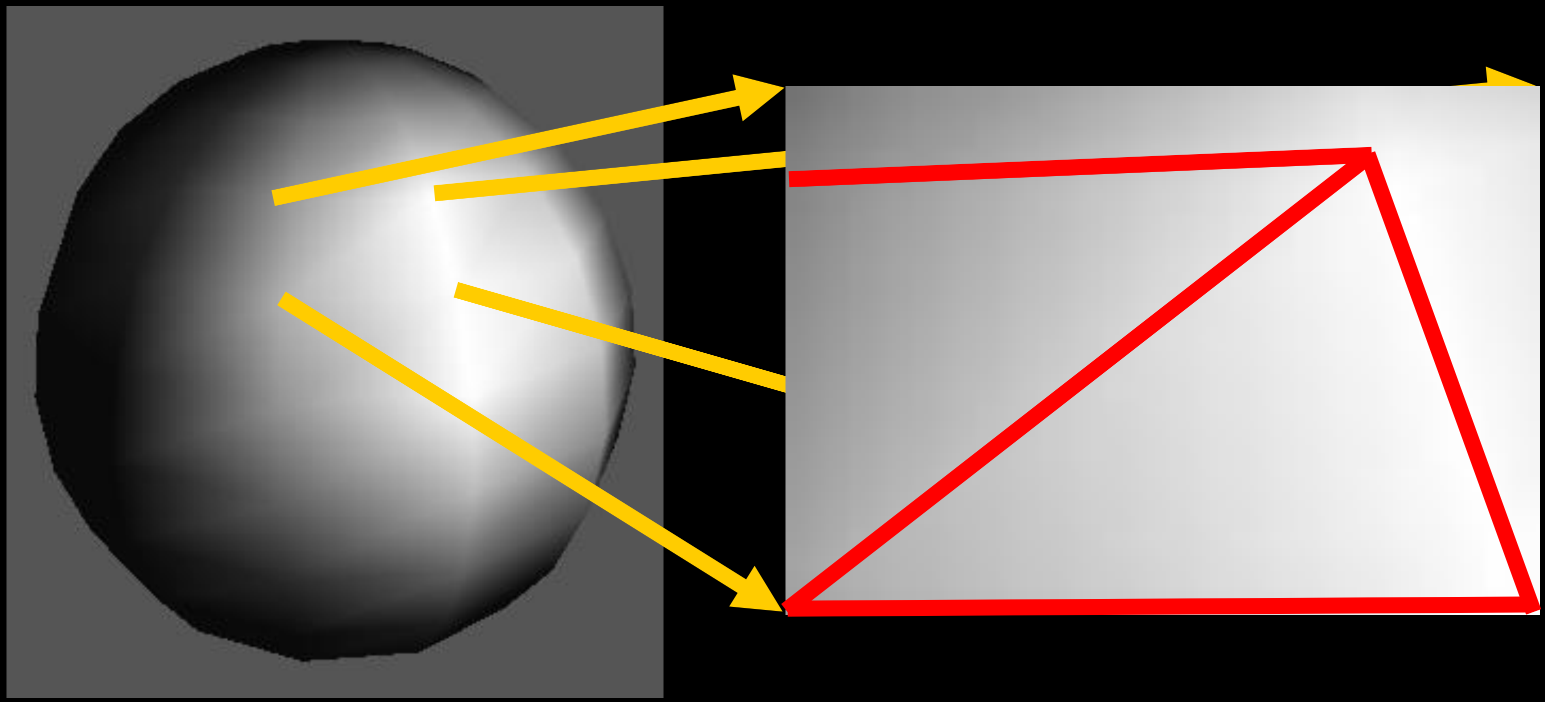
- Constant Shading
- Gouraud Shading
- Phong Shading

Constant Shading

- Constant Intensity or Flat Shading
- One color for the entire
- Fast
- Good for some objects
- What happens if object surface are small?
- Sudden intensity changes at borders

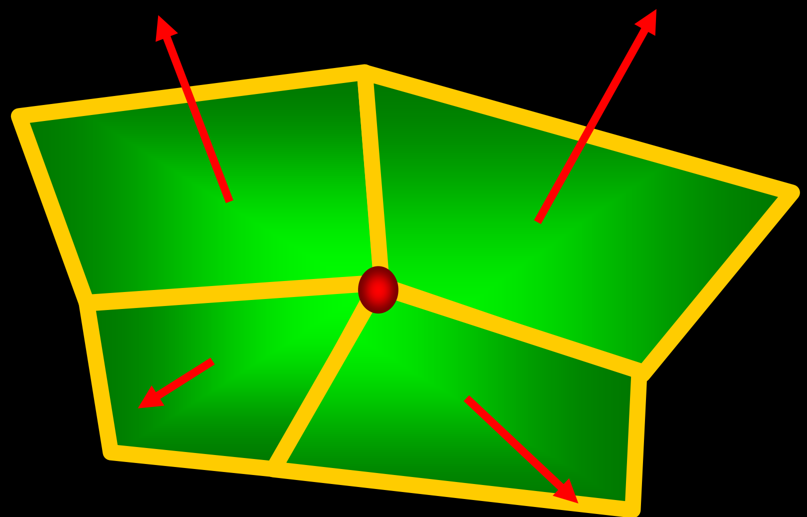
Gouraud Shading

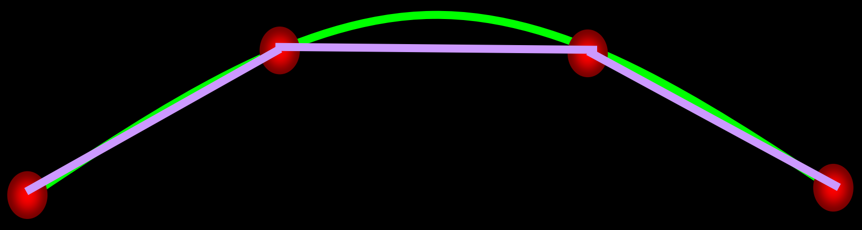
- Intensity Interpolation Shading
- Calculate lighting at the vertices. Then interpolate the colors as you scan convert



Gouraud Shading

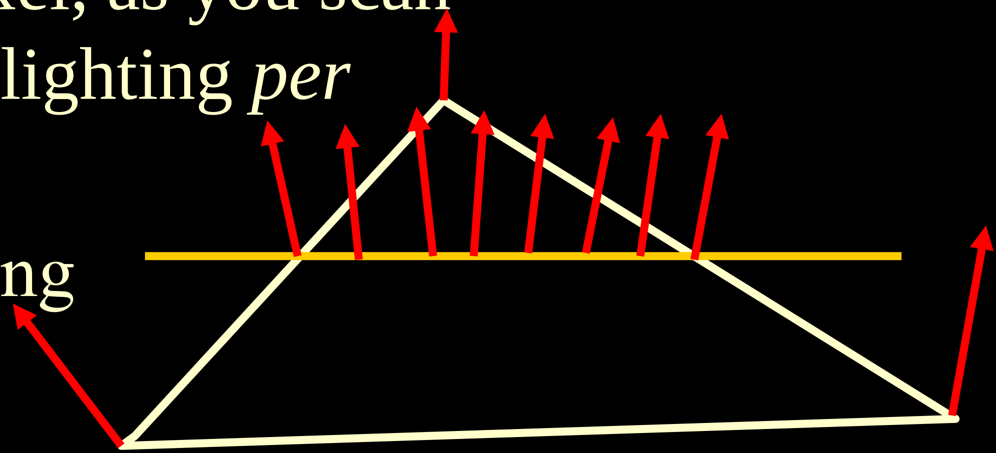
- Relatively fast, only do three calculations
- No sudden intensity changes





Phong Shading

- Interpolate the normal, since that is the information that represents the "curvature"
- Linearly interpolate the vertex normals. For ***each*** pixel, as you scan convert, *calculate* the lighting *per pixel*.
- True "per pixel" lighting
- Not done by most hardware/libraries/etc



Shading Techniques

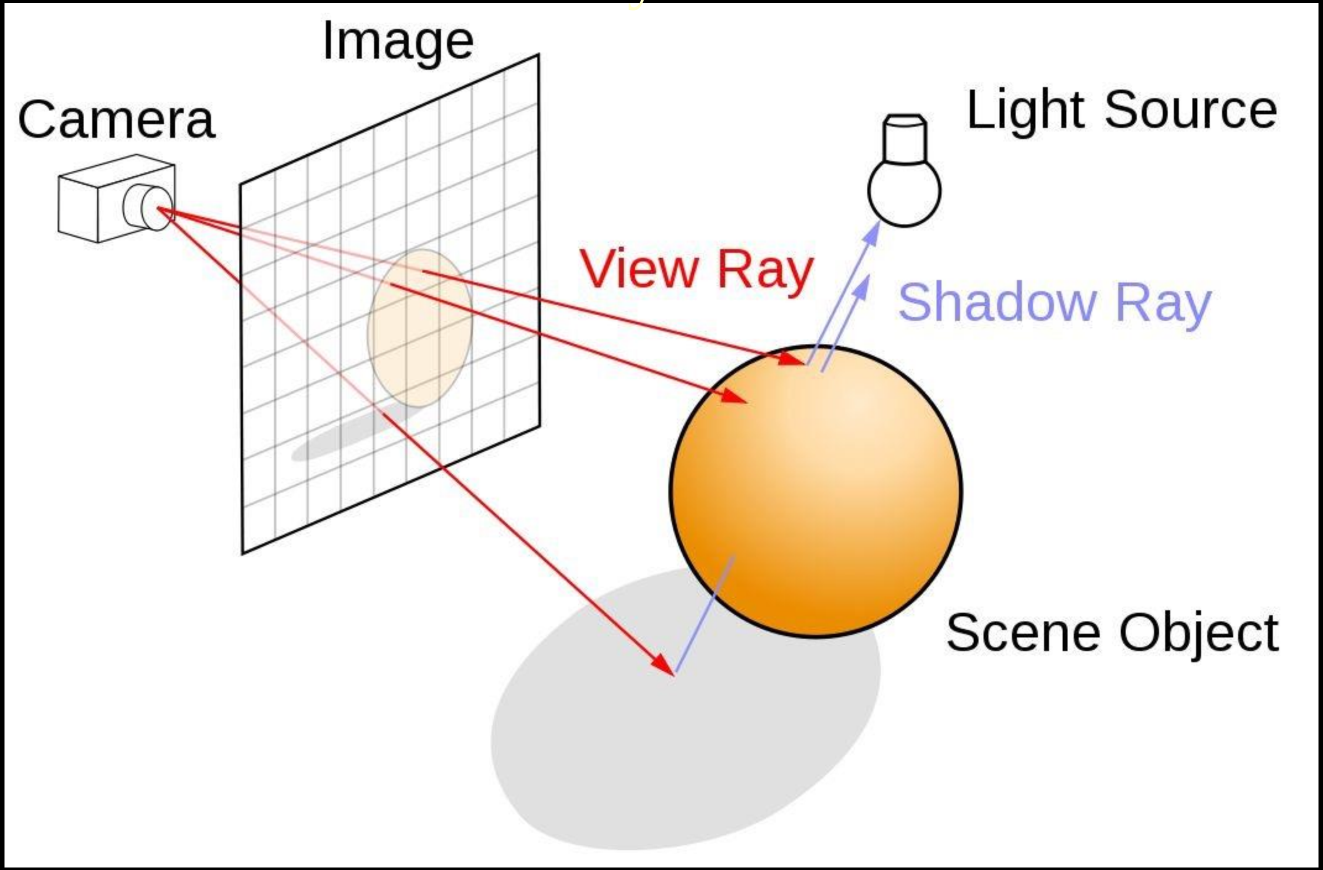
- Constant Shading
  - Calculate one lighting calculation (pick a vertex) per triangle
  - Color the *entire* triangle the same color
- Gouraud Shading
  - Calculate three lighting calculations (the vertices) per triangle
  - Linearly interpolate the colors as you scan convert
- Phong Shading
  - While you scan convert, linearly interpolate the normals.
  - With the interpolated normal at each pixel, calculate the lighting at each pixel

Shadows

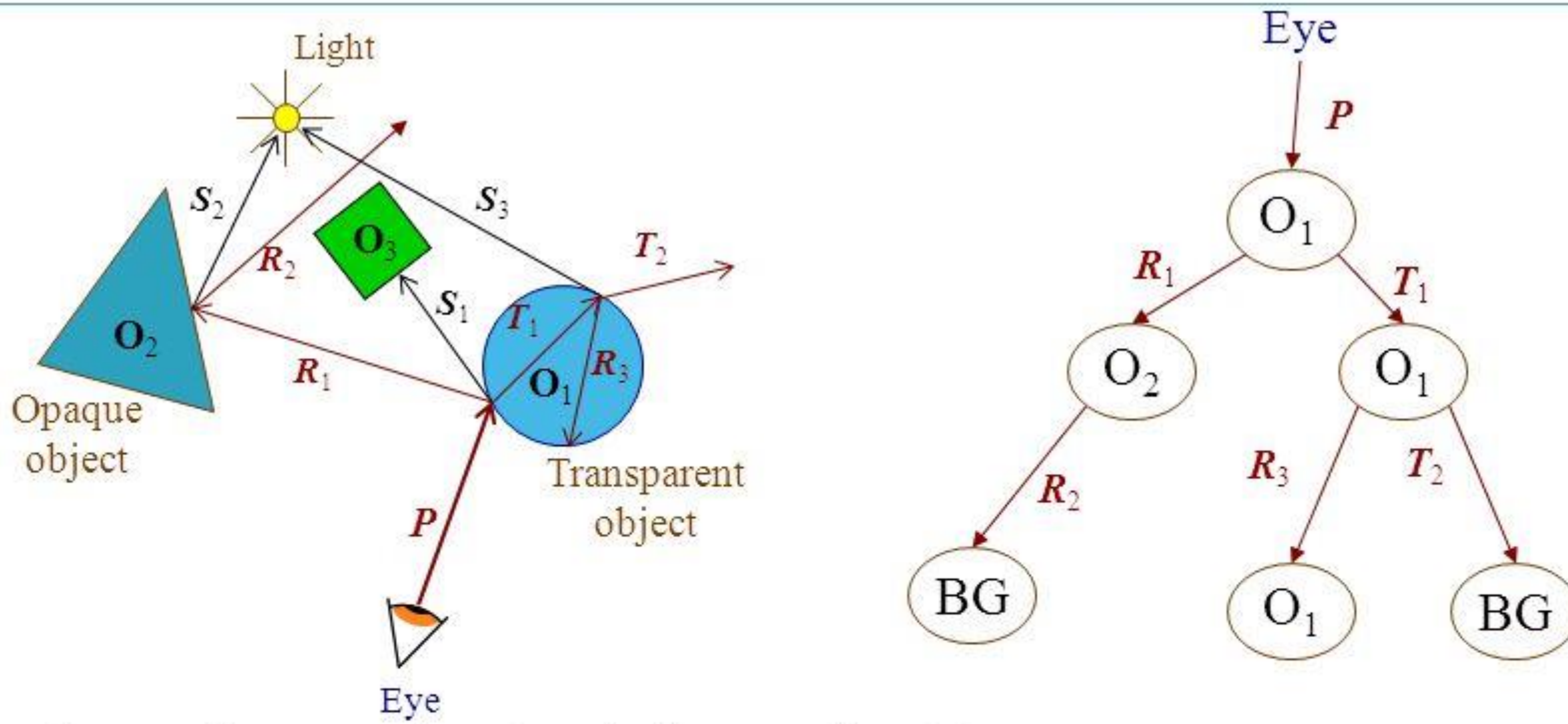
- Hidden surface methods used to locate the area where the light sources produce shadows
- Shadows – surface section cann’t see from the light source.

Ray tracing method

- Instead of finding visible surface for each pixel,
  Continue to bounce the ray around the scene



Ray Tracing: Ray Tree Example



Ray tree is evaluated from bottom up:

- Depth-first traversal
- The node colour is computed based on its children's colours.