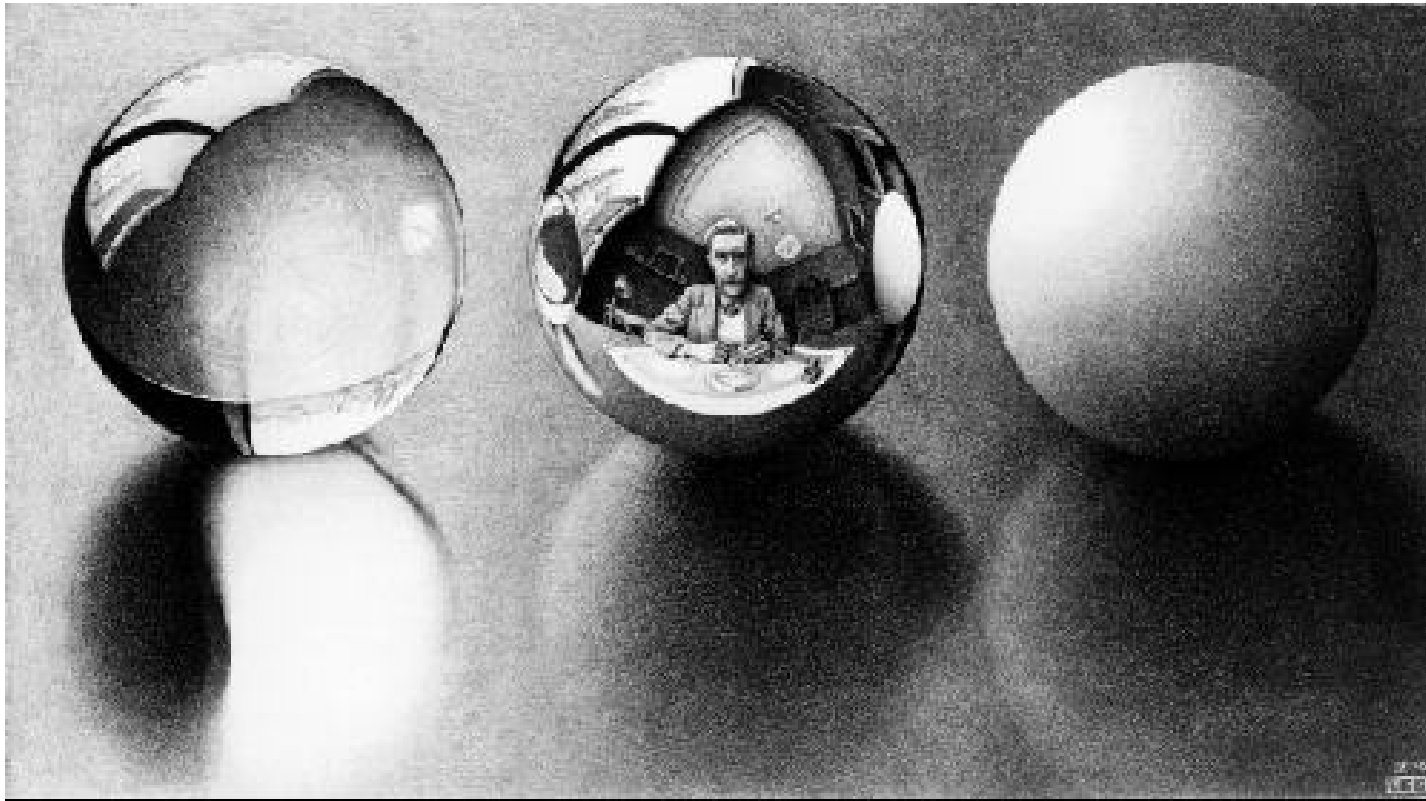


# Illumination Models and Shading

Foley & Van Dam, Chapter 16

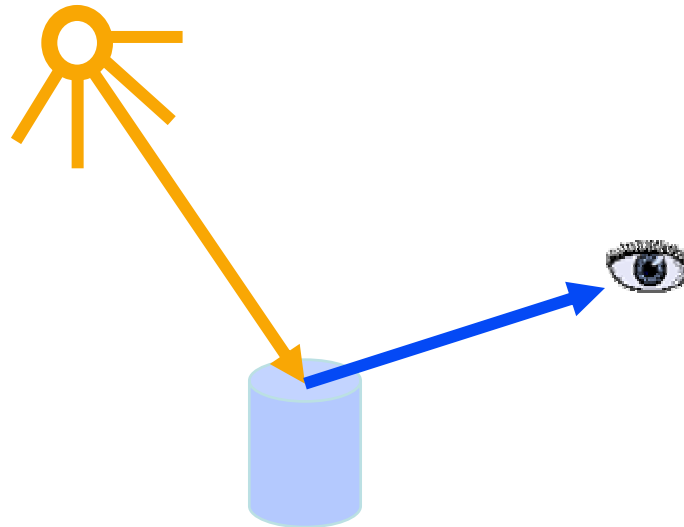


# Illumination Models and Shading

- Light Source Models
- Ambient Illumination
- Diffuse Reflection
- Specular Reflection
- Polygon Rendering Methods
- Flat Shading
- Gouraud Shading
- Phong Shading

# Illumination Models

- **Motivation:** In order to produce realistic images, we must simulate the appearance of surfaces under various lighting conditions
- **Illumination Model:** Given the illumination incident at a point on a surface, quantifies the reflected light

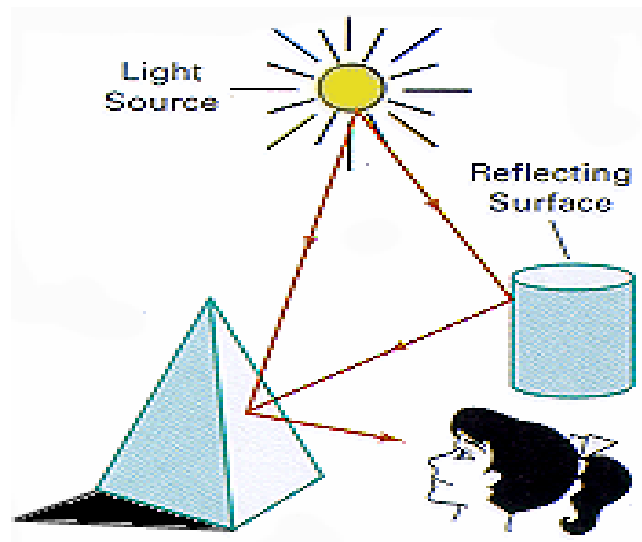


# Illumination Model Parameters

- Lighting effects are described with models that consider the interaction of light sources with object surfaces
- The factors determining the lighting effects are:
  - The **light source** parameters:
    - Positions
    - Electromagnetic Spectrum
    - Shape
  - The **surface** parameters
    - Position
    - Reflectance properties
    - Position of nearby surfaces
  - The **eye (camera)** parameters
    - Position
    - Sensor spectrum sensitivities

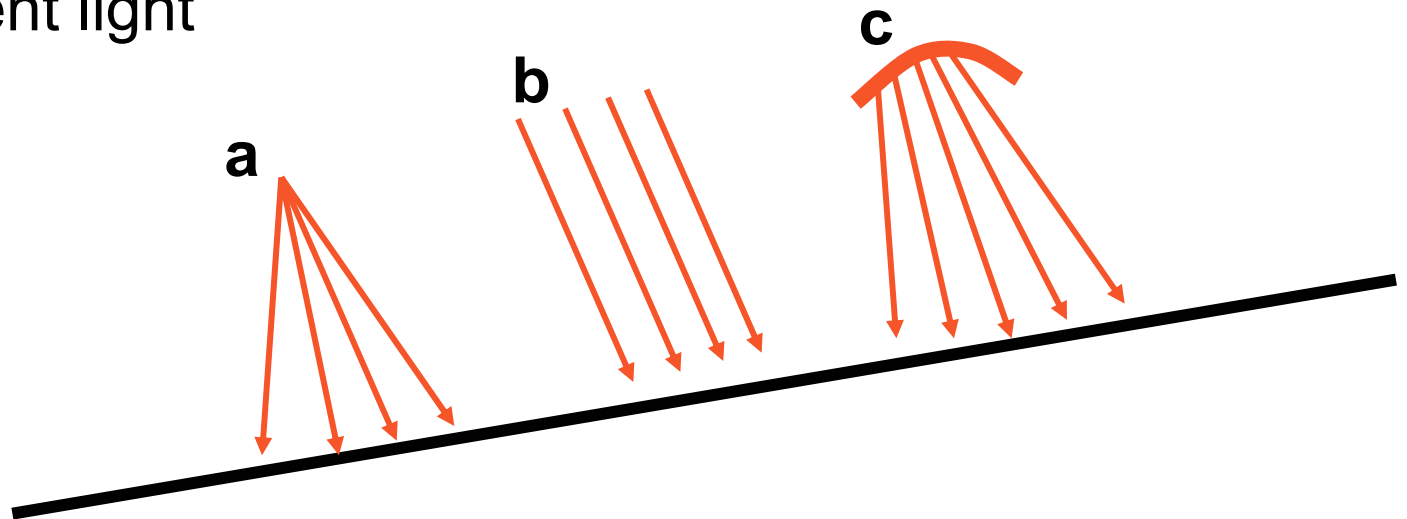
# Illumination Models and Rendering

- An **illumination model** is used to calculate the intensity of the light that is reflected at a given point on a surface
- A **rendering method** uses intensity calculations from the illumination model to determine the light intensity at all pixels in the image



# Light Source Models

- **Point Source** (a): All light rays originate at a point and radially diverge. A reasonable approximation for sources whose dimensions are small compared to the object size
- **Parallel source** (b): Light rays are all parallel. May be modeled as a point source at infinite distance (the sun)
- **Distributed source** (c): All light rays originate at a finite area in space. It models a nearby source, such as a fluorescent light



# Illumination Models

- Simplified and fast methods for calculating surfaces intensities, mostly empirical
- Calculations are based on optical properties of surfaces and the lighting conditions (no reflected sources nor shadows)
- Light sources are considered to be point sources
- Reasonably good approximation for most scenes

# Ambient Illumination

- Assume there is some non-directional light in the environment (background light)
- The amount of ambient light incident on each object is constant for all surfaces and over all directions
- Very simple model, not very realistic
- OpenGL default



# Ambient Illumination

- The reflected intensity  $I_{\text{amb}}$  of any point on the surface is:

$$I_{\text{amb}} = K_a I_a$$

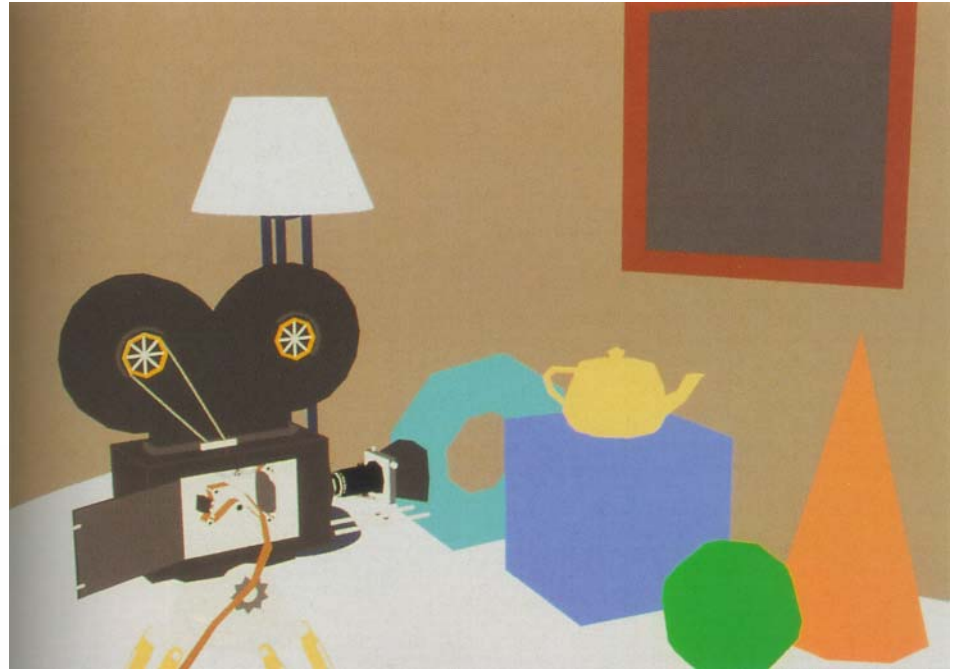
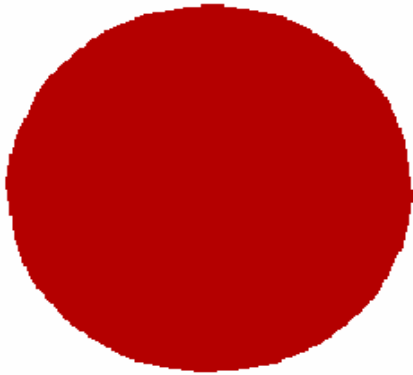
$I_a$  - ambient light intensity

$K_a \in [0, 1]$  - surface ambient reflectivity

- In principle  $I_a$  and  $K_a$  are functions of color, so we have  $I_{\text{amb}}^{\text{R}}$ ,  $I_{\text{amb}}^{\text{G}}$  and  $I_{\text{amb}}^{\text{B}}$

# Ambient Illumination

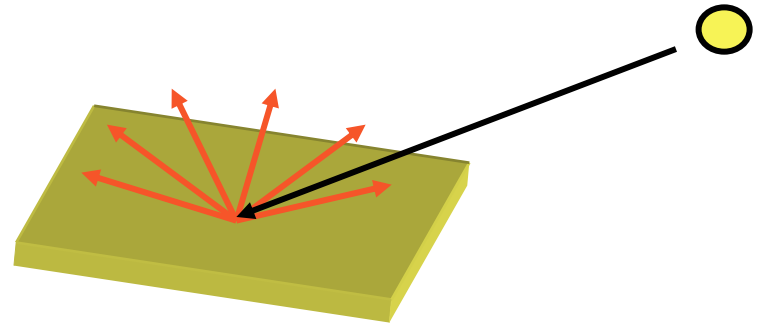
- **Example:**



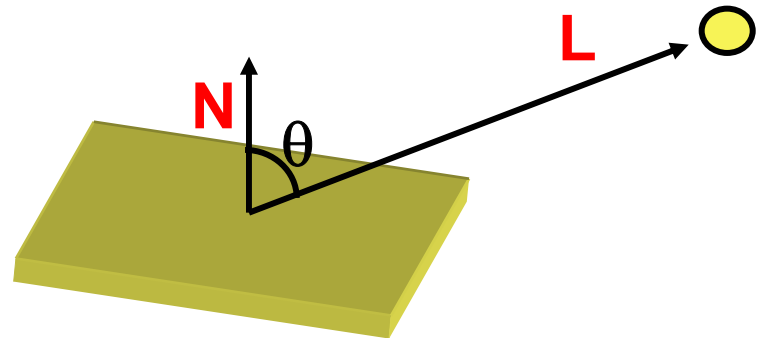
# Diffuse Reflection

- Diffuse (Lambertian) surfaces are rough or grainy, like clay, soil, fabric

- The surface appears equally bright from all viewing directions

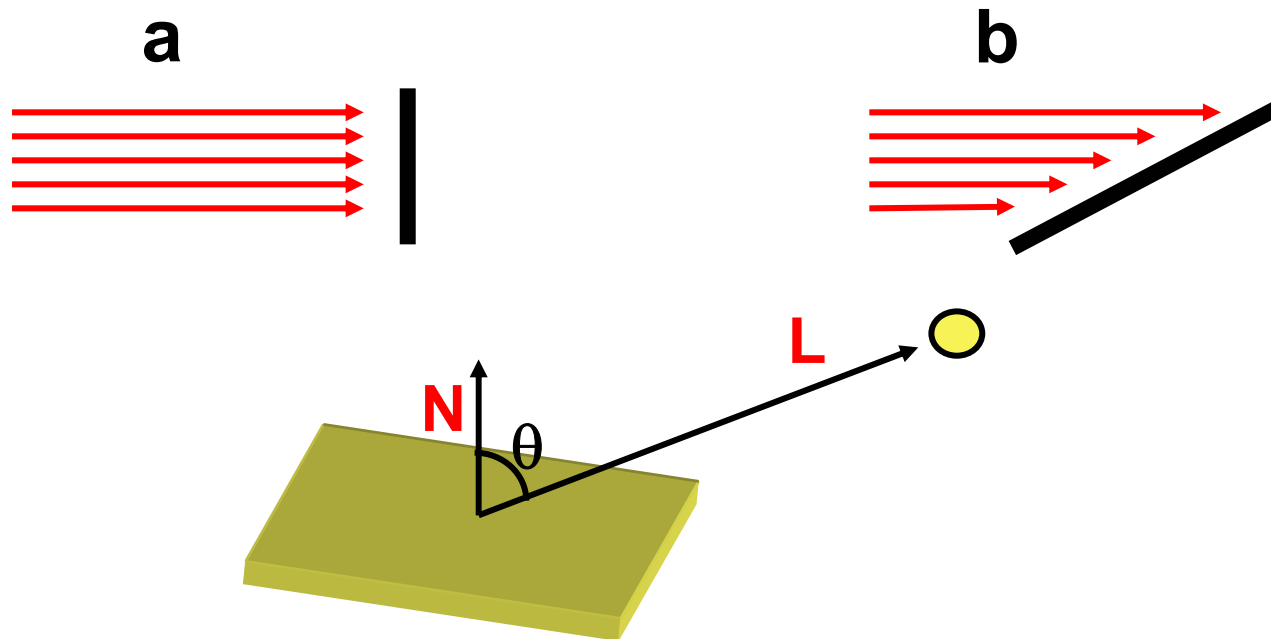


- The brightness at each point is proportional to  $\cos(\theta)$



# Diffuse Reflection

- Brightness is proportional to  $\cos(\theta)$  because a surface (a) perpendicular to the light direction is more illuminated than a surface (b) at an oblique angle



# Diffuse Reflection

- The reflected intensity  $I_{\text{diff}}$  of a point on the surface is:

$$I_{\text{diff}} = K_d I_p \cos(\theta) = K_d I_p (N \cdot L)$$

$I_p$  - the point light intensity. May appear as attenuated source  $f_{\text{att}}(r)I_p$

$K_d \in [0, 1]$  - the surface diffuse reflectivity

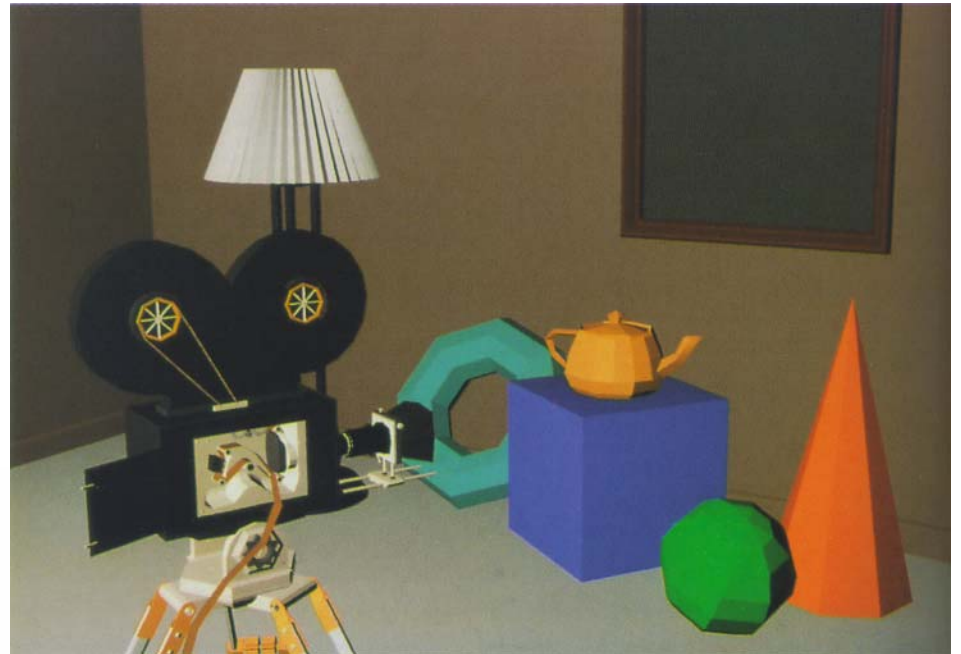
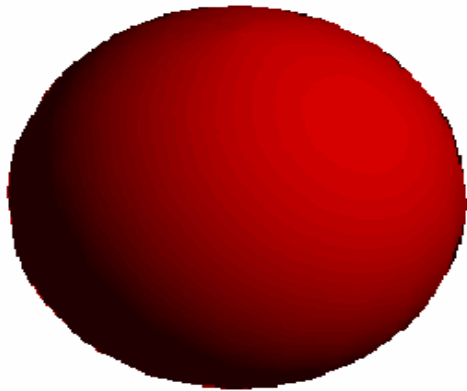
$N$  - the surface normal

$L$  - the light direction

NOTE: If  $N$  and  $L$  have unitary length:  $\cos(\theta) = N \cdot L$

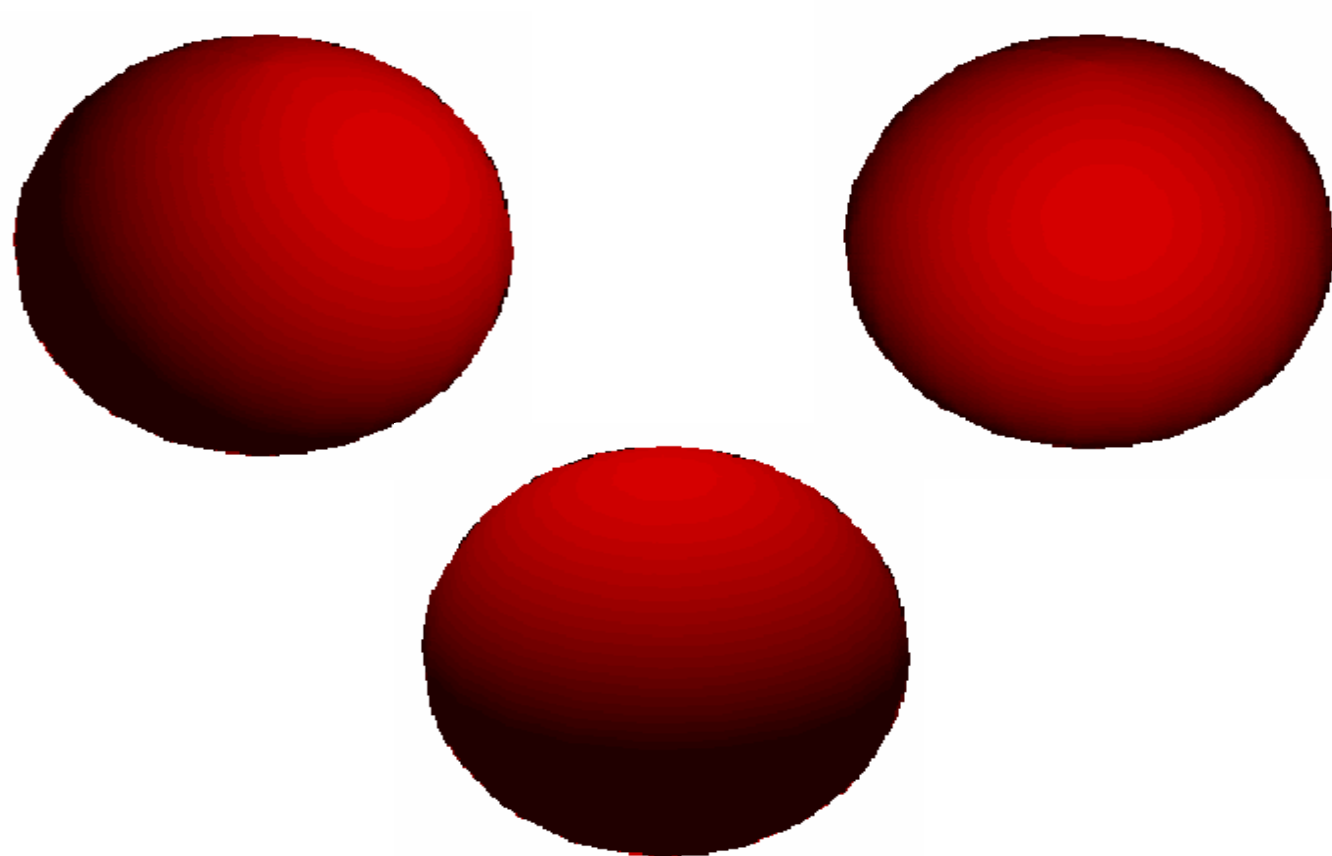
# Diffuse Reflection

- **Example:**



# Diffuse Reflection

- **Example:** diffuse reflection from different light directions



# Diffuse Reflection

- Commonly, there are two types of light sources:
  - A background ambient light
  - A point light source
- The equation that combines the two models is:

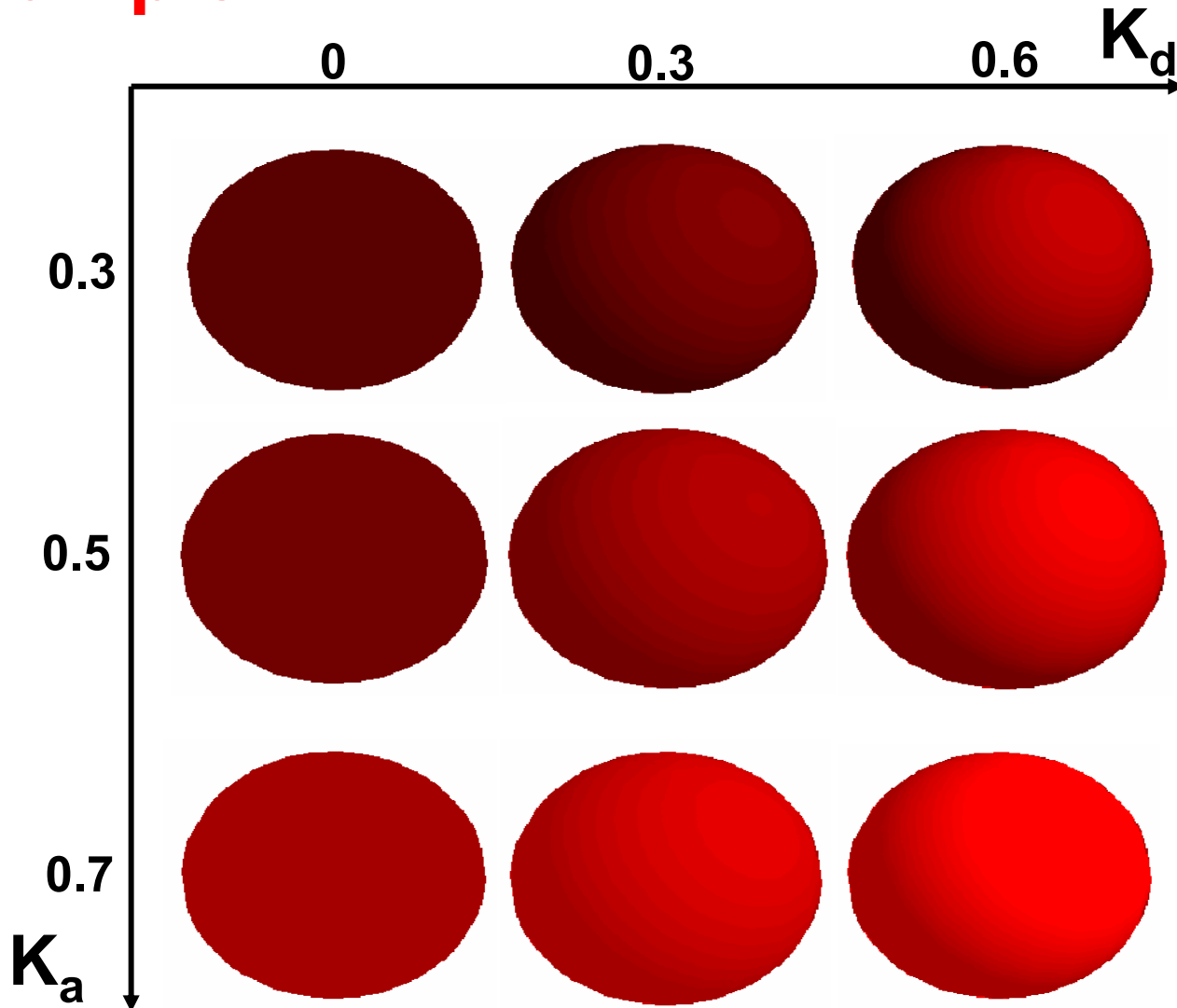
$$I = I_{\text{diff}} + I_{\text{amb}} = K_d I_p N \cdot L + K_a I_a$$

- Note this is the model for one color and it should be replicated for each channel:  $I^R$ ,  $I^G$  and  $I^B$



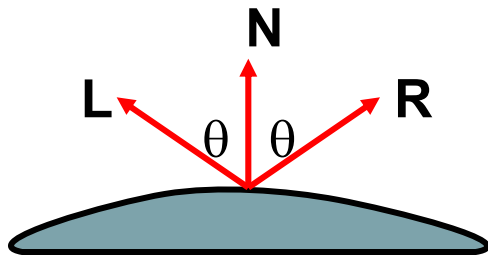
# Diffuse Reflection

- **Example:**

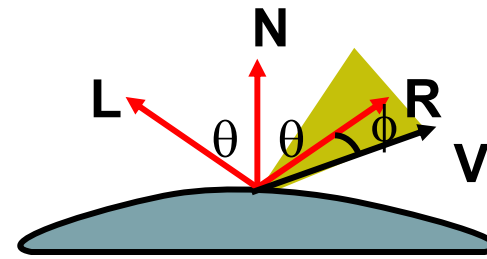


# Specular Reflection

- Models shiny and glossy surfaces (like metal, plastic, etc..) with **highlights**
- Reflectance intensity changes with reflected angle
- An ideal specular surface (mirror) reflects light exclusively in one direction: **R**
- Glossy objects are not ideal mirrors and reflect in the immediate vicinity of  $R$



Ideal specular surface



Non-ideal specular surface

# Specular Reflection

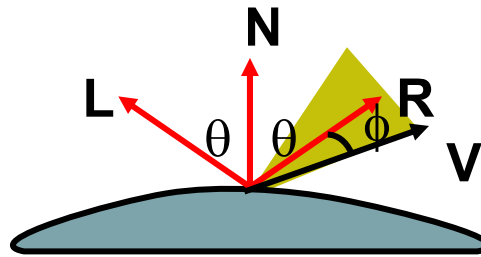
- **The Phong Model**: reflected specular intensity falls off as some power of  $\cos(\phi)$ :

$$I_{\text{spec}} = K_s I_p \cos^n(\phi) = K_s I_p (\mathbf{R} \cdot \mathbf{V})^n$$

$K_s$  - the surface specular reflectivity

$n$  – specular reflection parameter, determining the deviation from ideal specular surface

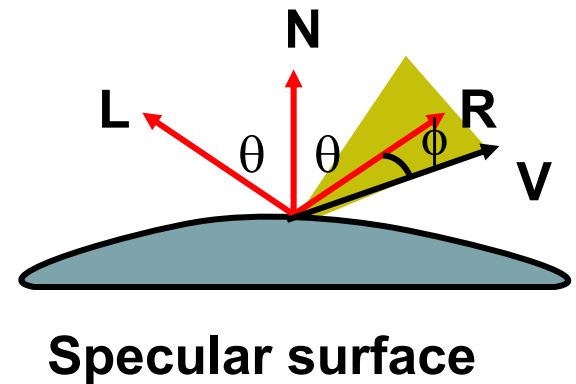
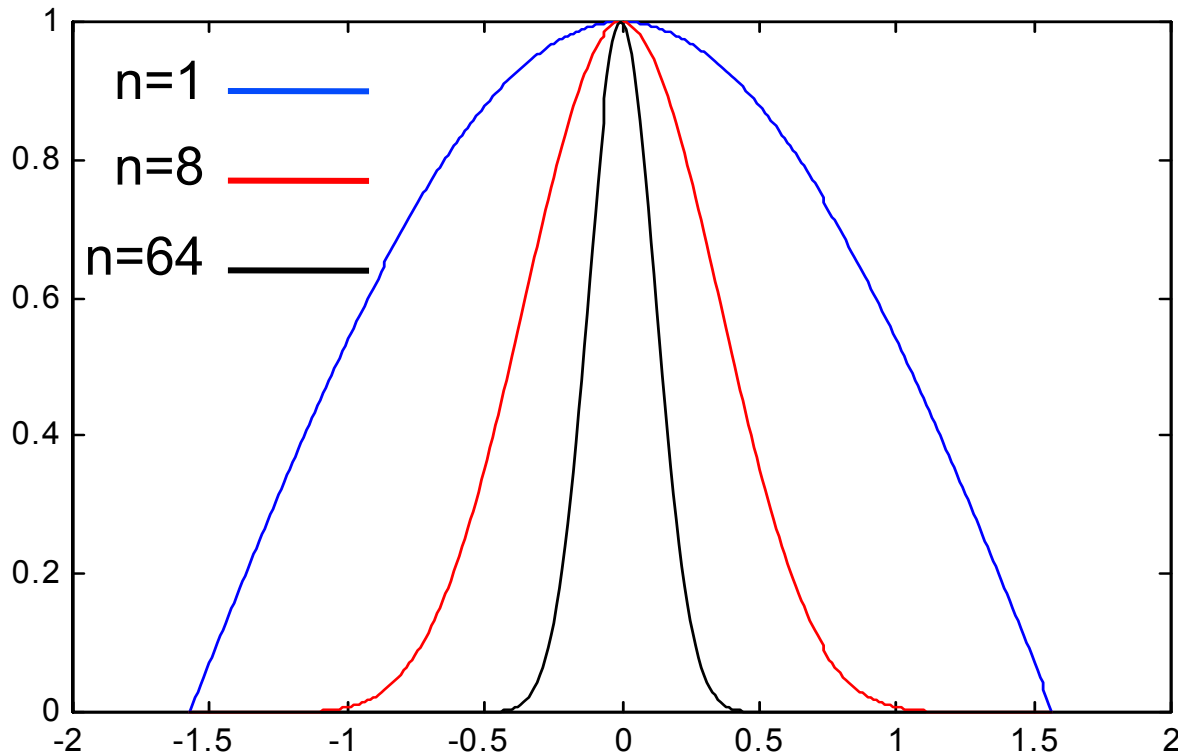
(for a perfect mirror  $n = \infty$ )



Specular surface

# Specular Reflection

- **The Phong Model:** plots of  $\cos^n(\phi)$  for three values of the specular parameter  $n$



# Specular Reflection

- The illumination equation combined with diffuse reflection is:

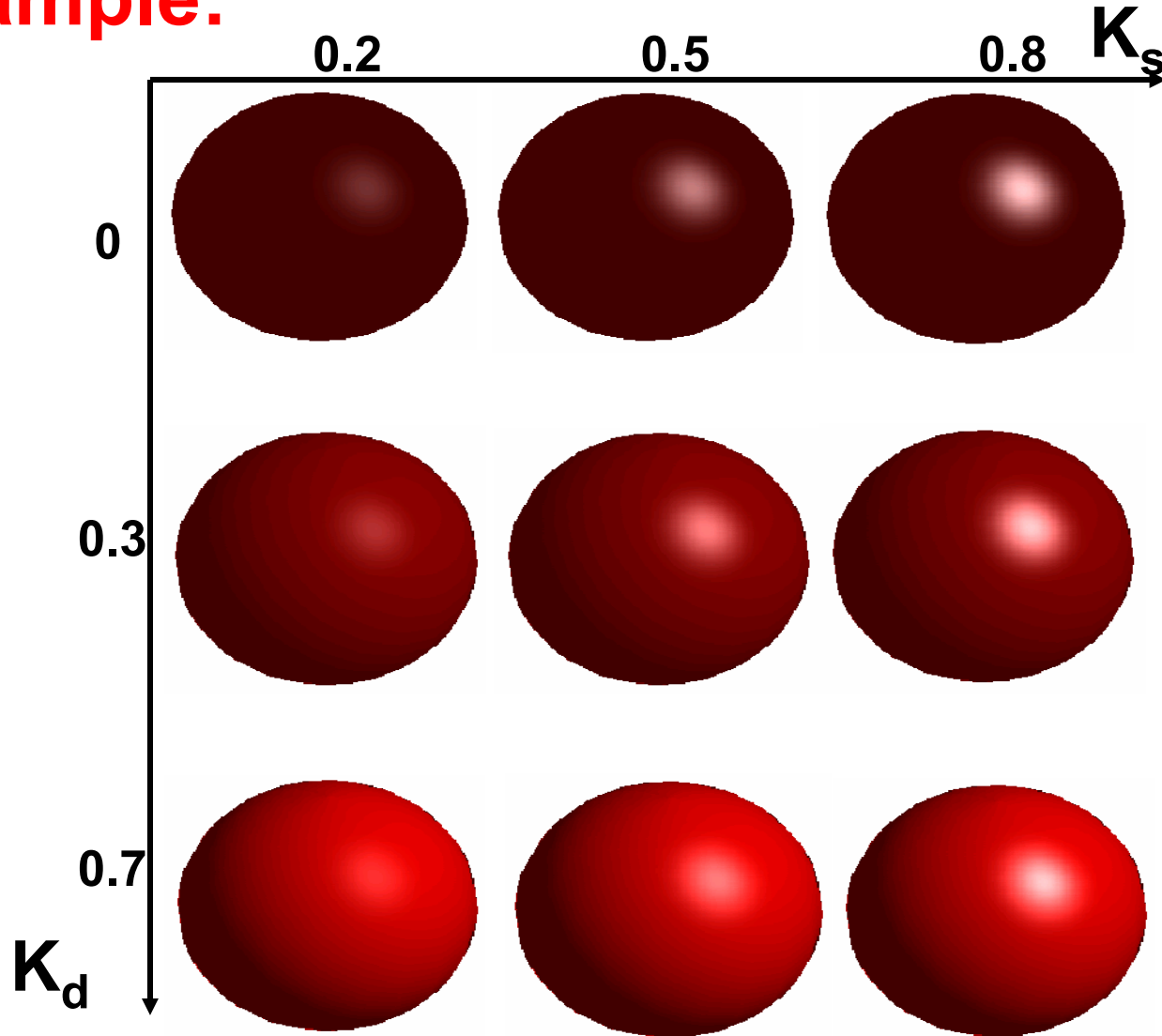
$$I = I_{amb} + I_{diff} + I_{spec} = K_a I_a + I_p (K_d N \cdot L + K_s (R \cdot V)^n)$$

- If **k** light sources are present in the scene:

$$I = I_{amb} + \sum_k (I_{diff}^k + I_{spec}^k)$$

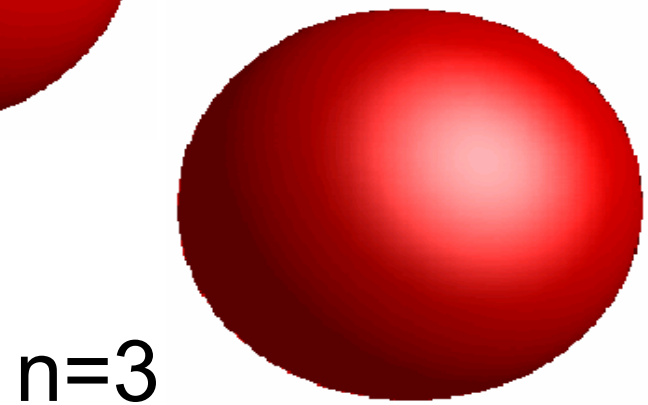
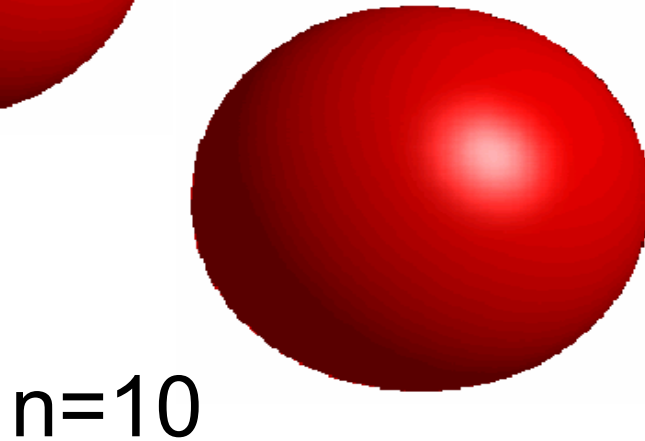
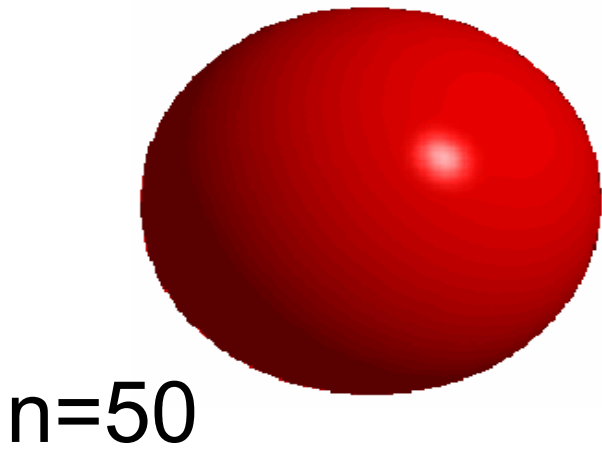
# Specular Reflection

- **Example:**



# Specular Reflection

- **Example:** effects of the specular parameter  $n$

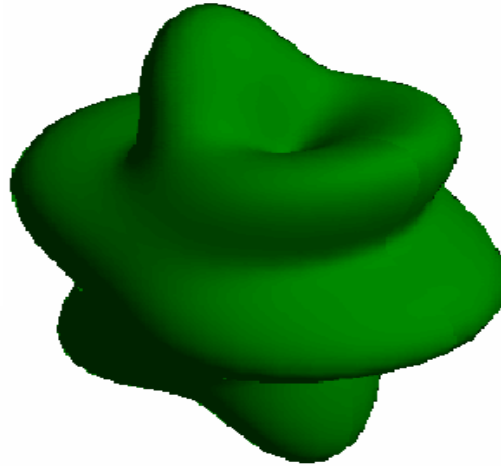


# Specular Reflection

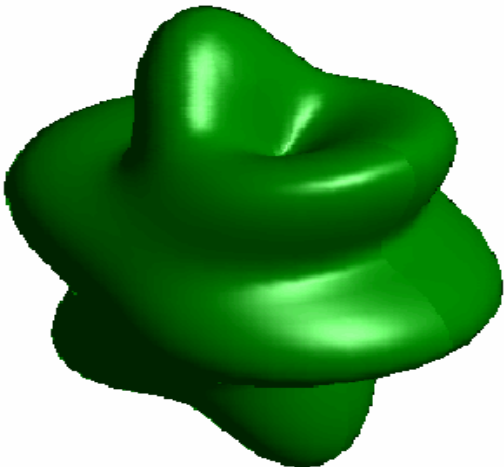
- **Example:**



Ambient Illumination



Ambient + Diffuse

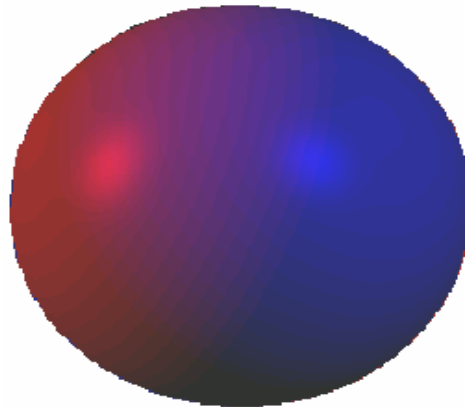
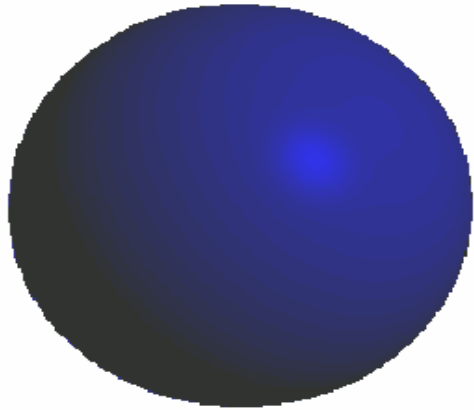


Ambient + Diffuse + Specular



# Composing Light Sources

- **Example:**



# Polygon Rendering Methods

- A freeform surface can be approximated by polyhedra
- **Rendering:** calculate the illumination at each surface point
- Applying the illumination model at each surface point is computationally expensive

# Flat Shading

- A single intensity is calculated for each surface polygon
- Fast and simple method
- Gives reasonable result only if all of the following assumptions are valid:
  - The object is a polyhedron
  - Light source is far away from the surface so that  $N \cdot L$  is constant over each polygon
  - Viewing position is far away from the surface so that  $V \cdot R$  is constant over each polygon



# Gouraud Shading

- Renders the polygon surface by linearly interpolating intensity values across the surface

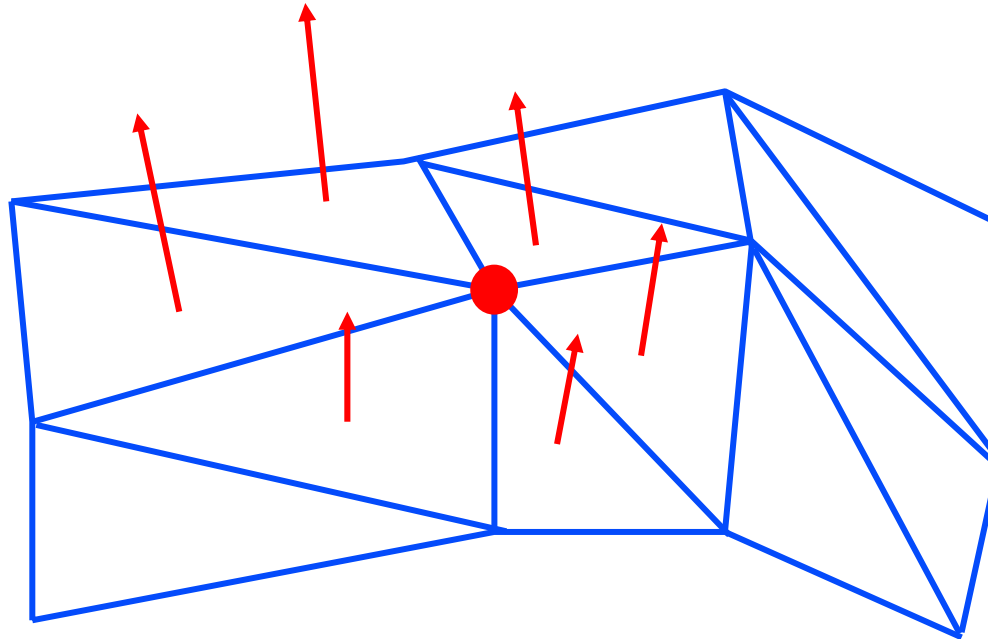
## Gouraud Shading Algorithm:

1. Determine the normal at each polygon vertex
2. Apply an illumination model to each vertex to calculate the vertex intensity
3. Linearly interpolate the vertex intensities over the surface polygon

# Gouraud Shading

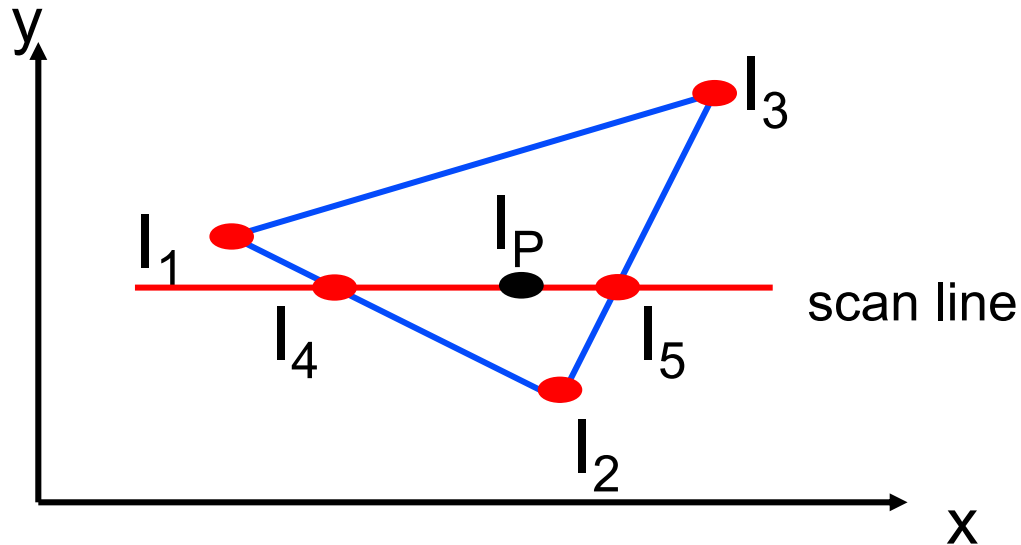
- The normal  $N_v$  of a vertex is an average of all neighboring normals:

$$N_v = \frac{\sum_k N_k}{\left| \sum_k N_k \right|}$$



# Gouraud Shading

- Interpolation of the vertex intensities



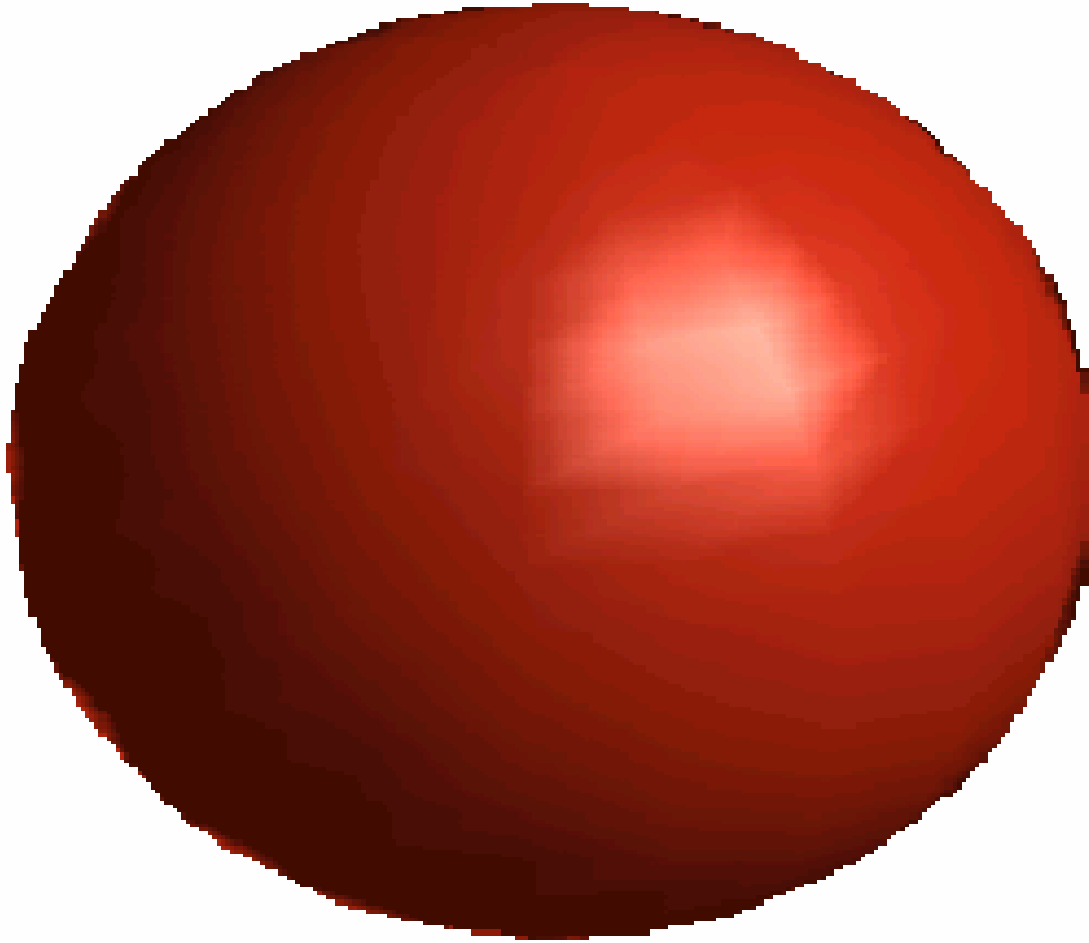
$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

$$I_5 = \frac{y_5 - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_5}{y_3 - y_2} I_2$$

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

# Gouraud Shading

- **Example:** Gouraud shading of a sphere



# Phong Shading

- A more accurate method for rendering a polygon surface is to interpolate normal vectors, and then apply the illumination model to each surface point

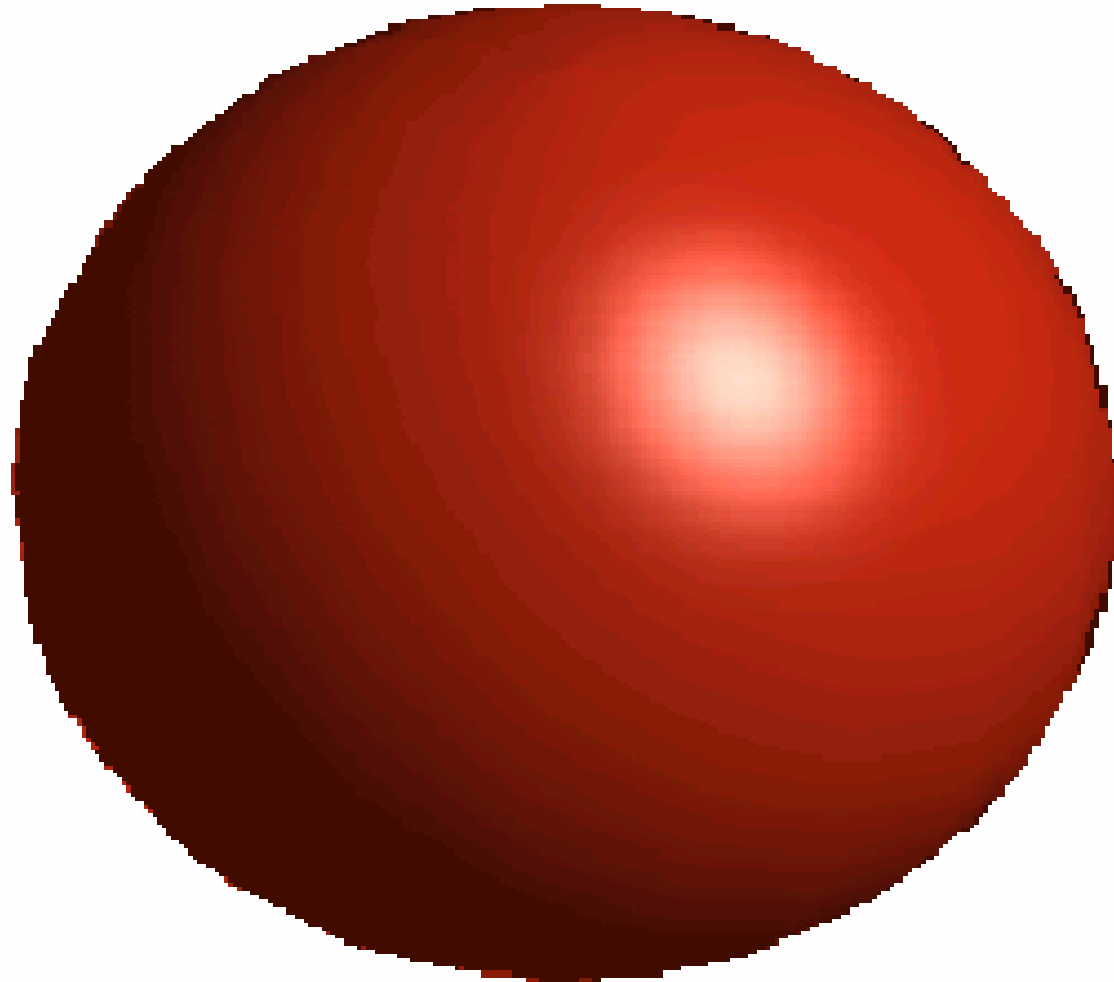
## Phong Shading Algorithm:

1. Determine the normal at each polygon vertex
2. Linearly interpolate the vertex normals over the surface polygon
3. Apply the illumination model along each scan line to calculate intensity of each surface point



# Phong Shading

- **Example:** Phong shading of a sphere

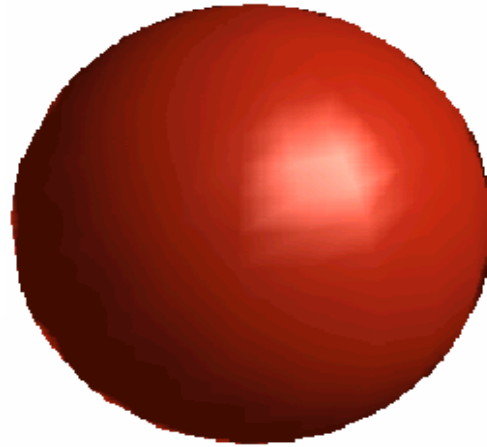


# Polygon Rendering Methods

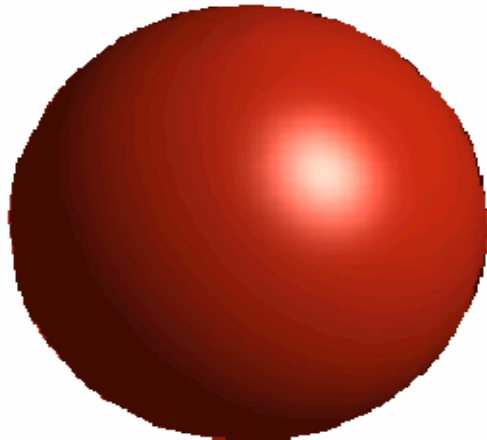
- **Example:**



Flat



Gouraud



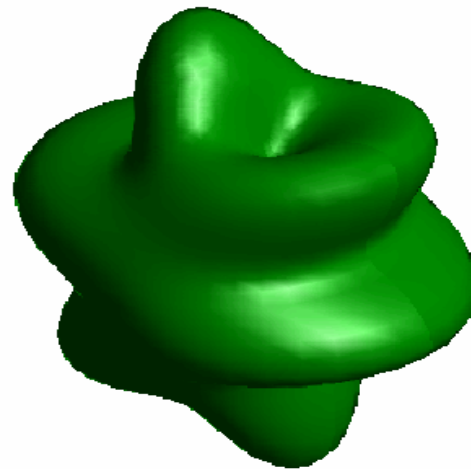
Phong

# Polygon Rendering Methods

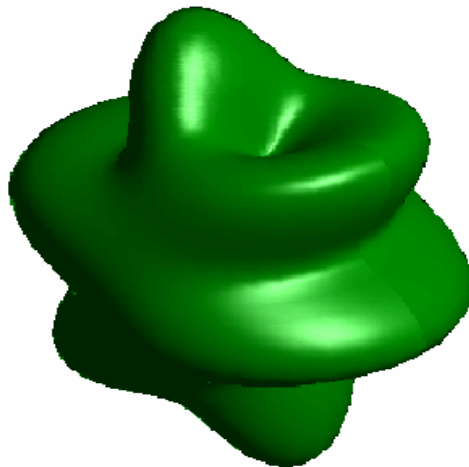
- **Example:**



Flat



Gouraud



Phong

# Polygon Rendering Methods

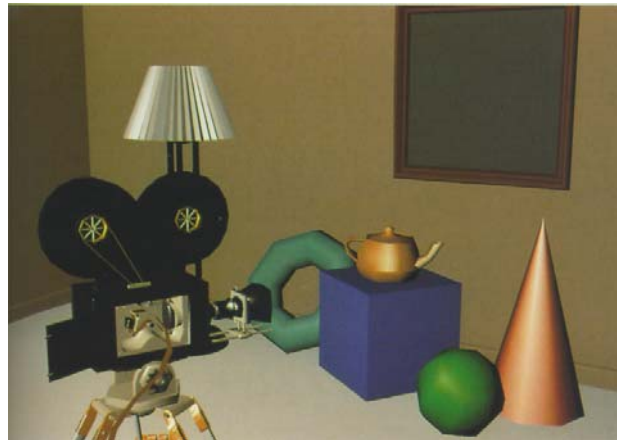
- **Example:**



Flat



Gouraud



Phong