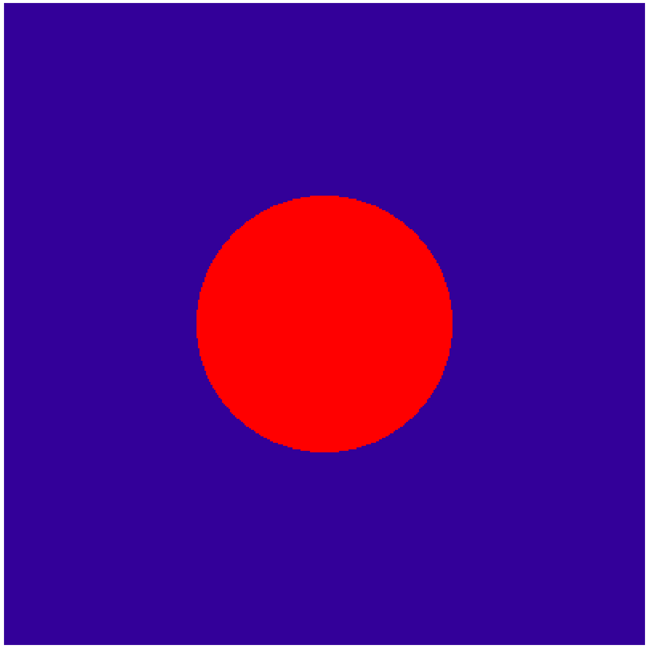
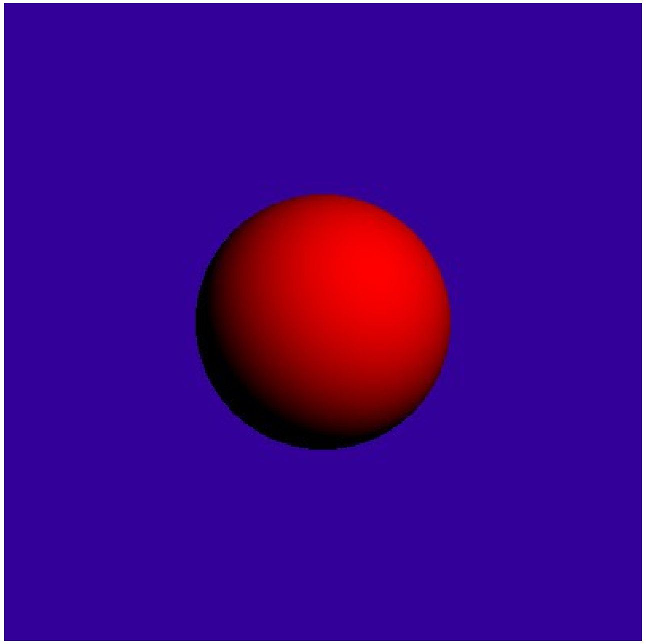
Computer Graphics

Illumination Models &
Surface Rendering Methods

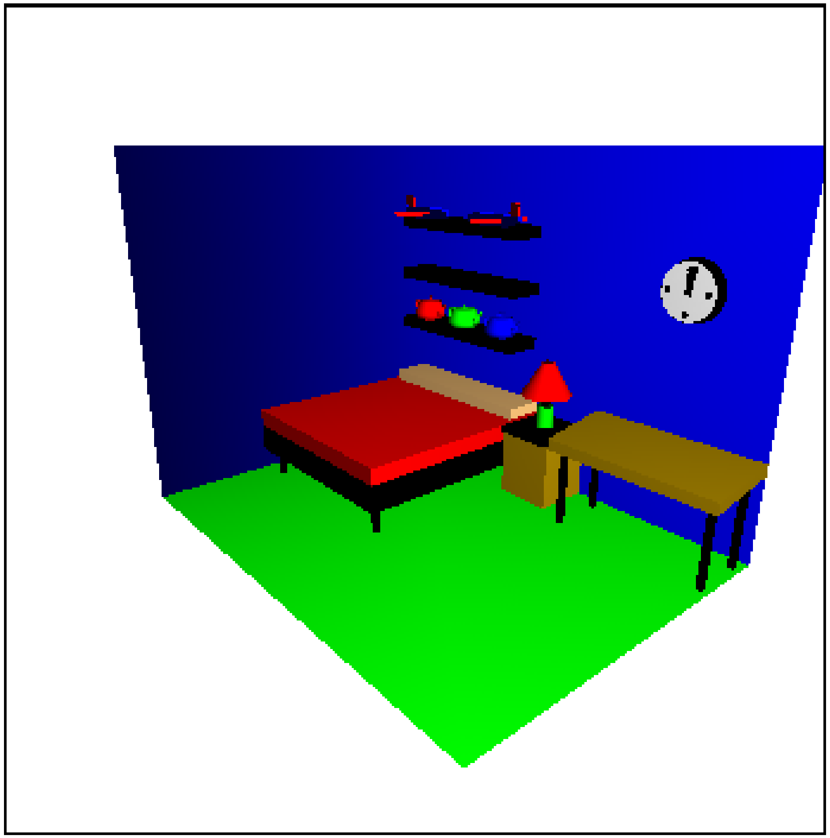
Why Lighting?

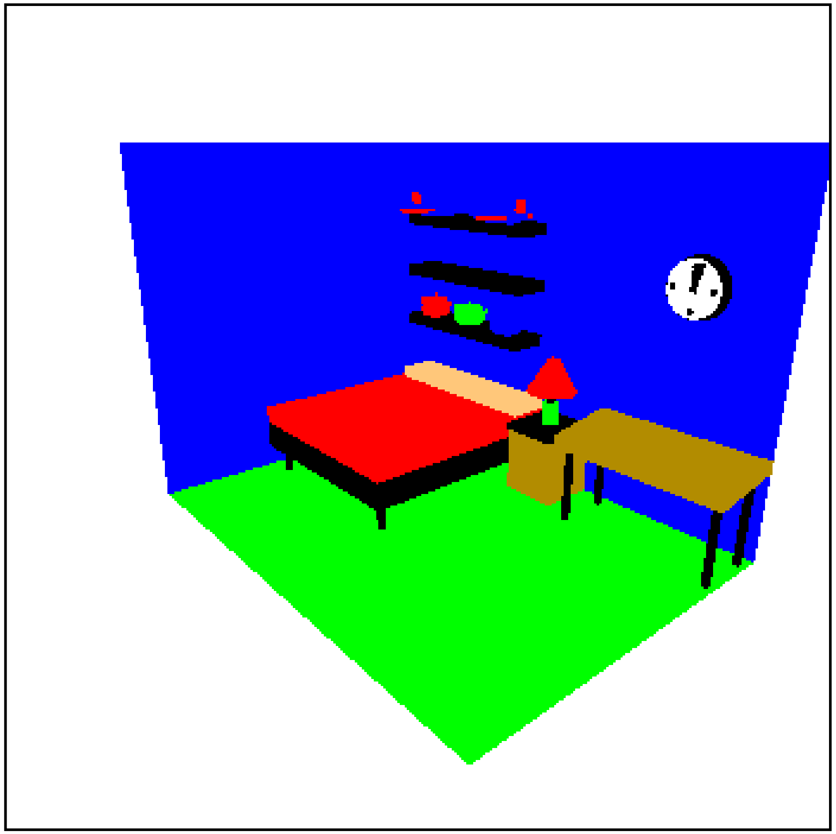
- If we don't have lighting effects nothing looks three dimensional!





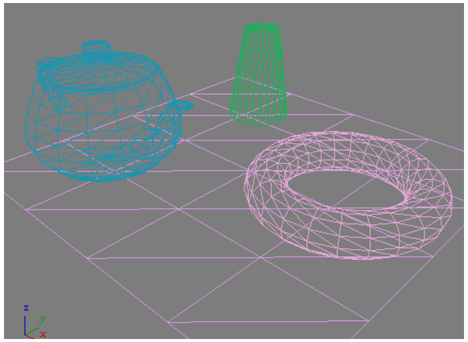
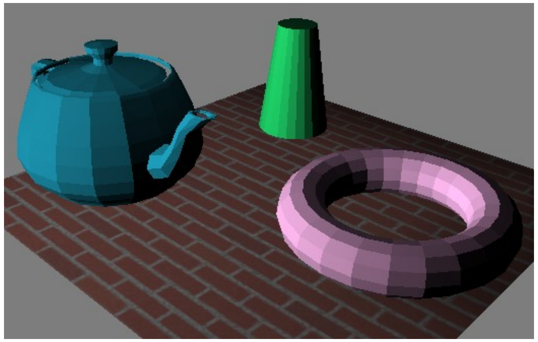
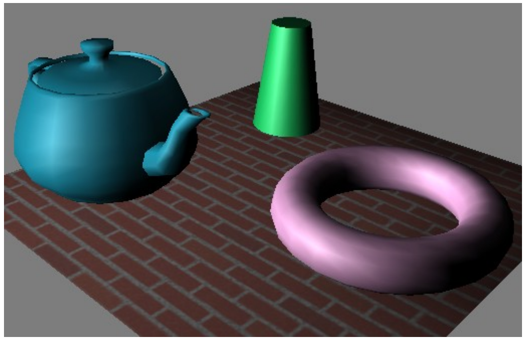
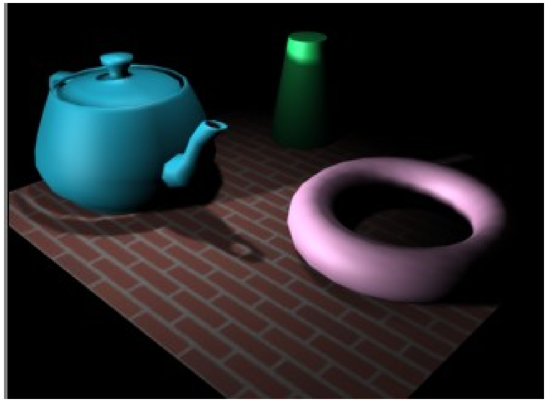
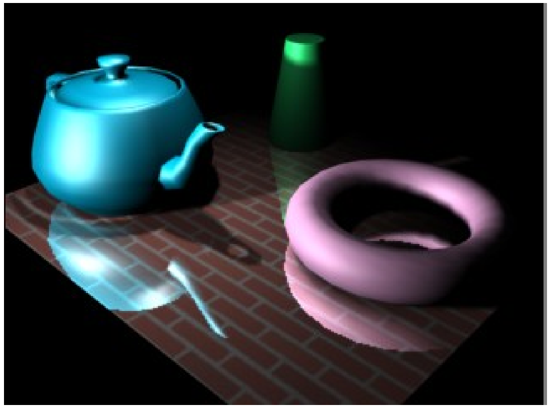
Why Lighting? (cont…)





Introduction

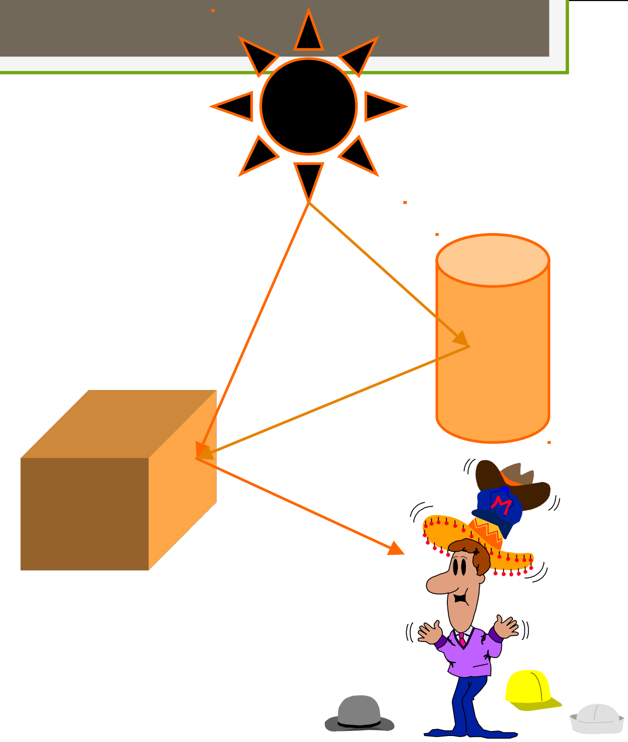
- Realistic displays of a scene
  - Perspective projections of objects
  - Applying lighting effects

Introduction (Contd.)

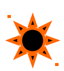
- Illumination model
  - Lighting model or Shading model
  - Calculate the <u>intensity of light</u> for a given point on the surface of an object
- Surface-Rendering algorithm
  - Use the intensity of a given point to determine the light intensity for all projected pixel position in a polygon

Light Sources



- Types of light source
  - light source (direct)
  - light reflector (indirect)
- Two light emitter models
  - Point light source, see (b)
  - Distributed (Area) light source, see (a)

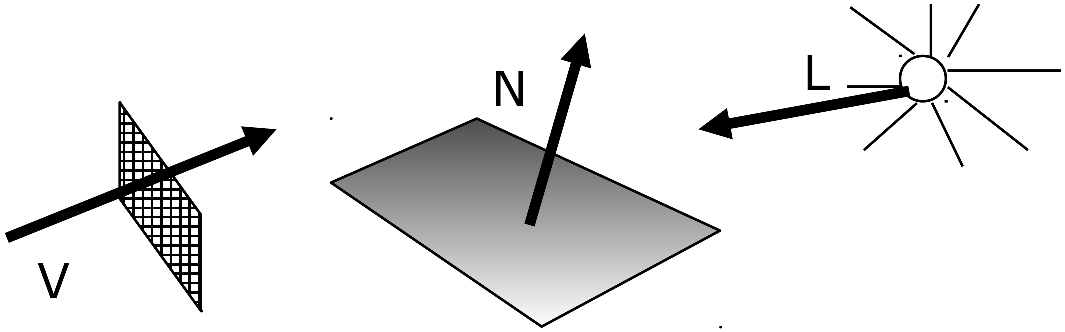
(a)



(b)

Illumination Models

- Concerning methods for calculating light intensity
  - Also called *Lighting* Models
  - An approximation for physical optical laws

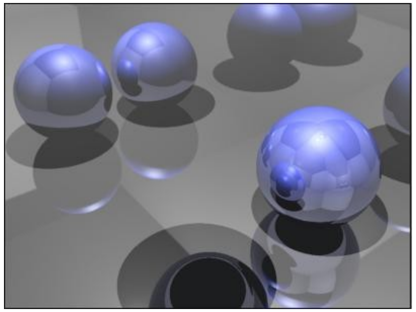


- Position
- Orientation
- Material
- Light source
- Viewer

Types of illumination models

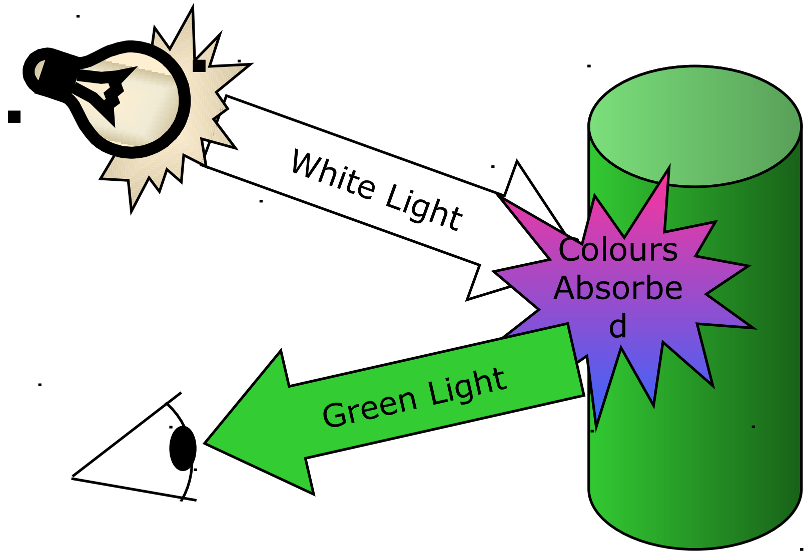
- Local Illumination Models
  - Only considering the interchanges of the light sources

- Global Illumination Models
  - Concerning the interchange of light between all surfaces
  - Ray-Tracing, Light as a particle
  - Radiosity, Light as a energy



Reflected Light

- The colours that we perceive are determined by the nature of the light reflected from an object
- For example, if white light is shone onto a green object most wavelengths are absorbed, while green light is reflected from the object



Background

- Illumination model (Shading Model) is used to Calculate intensity of light that we see at a given point on the surface of an object
- Surface Rendering algorithm uses the intensity calculation from an illumination model to determine the light intensity of all pixel positions for various surfaces in the scene
- Photorealism in computer graphics involves two elements:
  - Accurate graphical representations of object
  - Good physical descriptions of lighting effect in scene
- Illumination model: derived from physical laws that describe surface light intensities

Basic Illumination Model

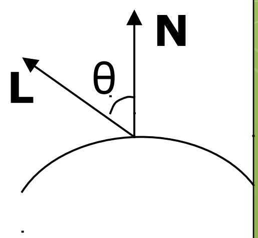
- Ambient Light → Background Light
  - Object not exposed directly to a light source is visible with ambient light
  - Has no spatial or directional characteristics
  - Amount of ambient light incident on each object is a constant for all surfaces and over all directions
  - Level for the ambient light in scene by parameter I_a
  - Each surface in the scene is illuminated with the constant intensity level I_a
  - The intensity of reflected light (intensity of illumination) depends upon optical properties of the surface
  - Ambient light produces flat shading → not desirable in general, so scenes are illuminated with other light source together with ambient light

Basic Illumination Model

- Diffuse Reflection
  - Reflected light intensity are constant over each surface in a scene independent of viewing direction → *ideal diffuse reflectors*
  - the diffuse-reflection coefficient (diffuse reflectivity) → k_d
    - Sets the fraction of light intensity that is reflected from each surface
    - k_d is a the function of surface color, but for our purpose we assume k_d to be constant
  - Diffuse reflection intensity when scene illuminated only with ambient light:

    $I_{ambdiff} = k_d I_a$

  - Lambertian reflectors
    - Follow lambart's cosine law → radiant energy from a small surface area *dA* is proportional to the cosine of angle θ between surface normal and incident light direction
      - For a point source with intensity I_l, the diffuse reflection intensity is
        $I_{l,diff} = k_d I_l \cos\theta$
        $I_{l,diff} = k_d I_l (\mathbf{N}.\mathbf{L})$
    Where **N** and **L** are unit vectors

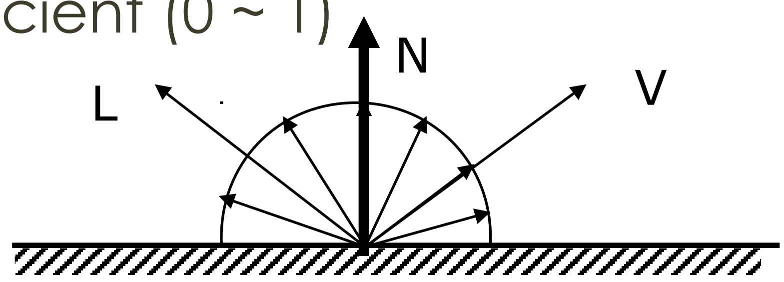


Diffuse Reflection

- Dull and matte surfaces exhibit diffuse reflection
  - Equally bright from all viewing angles
- The intensity on a given surface depends on the angle $\boldsymbol{\theta}$ between the light's direction ***L*** and surface's normal *N*

$$I = I_a K_a + I_p K_d \cos\theta$$

  - I_p : Intensity of point light source
  - K_d : diffuse-reflection coefficient (0 ~ 1)
  - $\cos\theta$: max($\cos\theta$, 0)

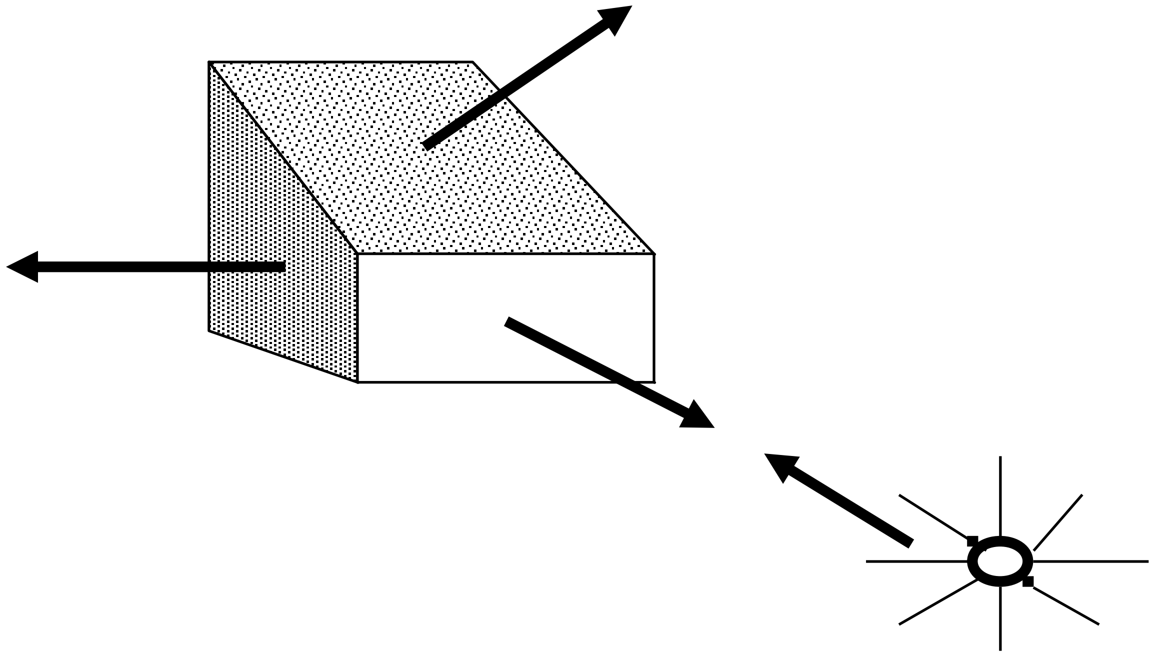


Total Diffuse Reflection

- Assuming that N and L have been normalized,

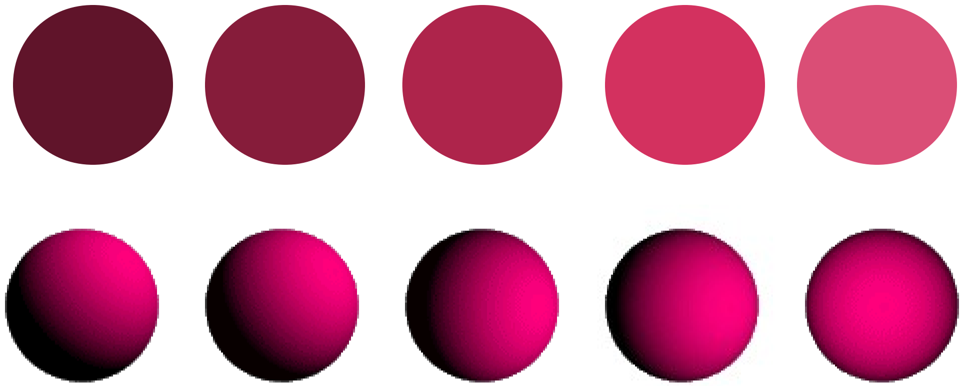
$$I = I_a K_a + I_p K_d (N \text{ dot } L)$$

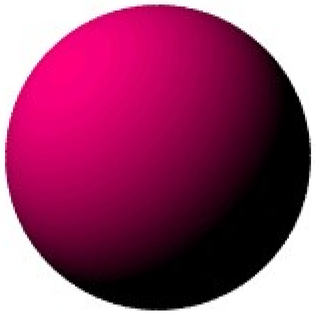
L is a constant if a point light source is at infinite (Called directional light source)



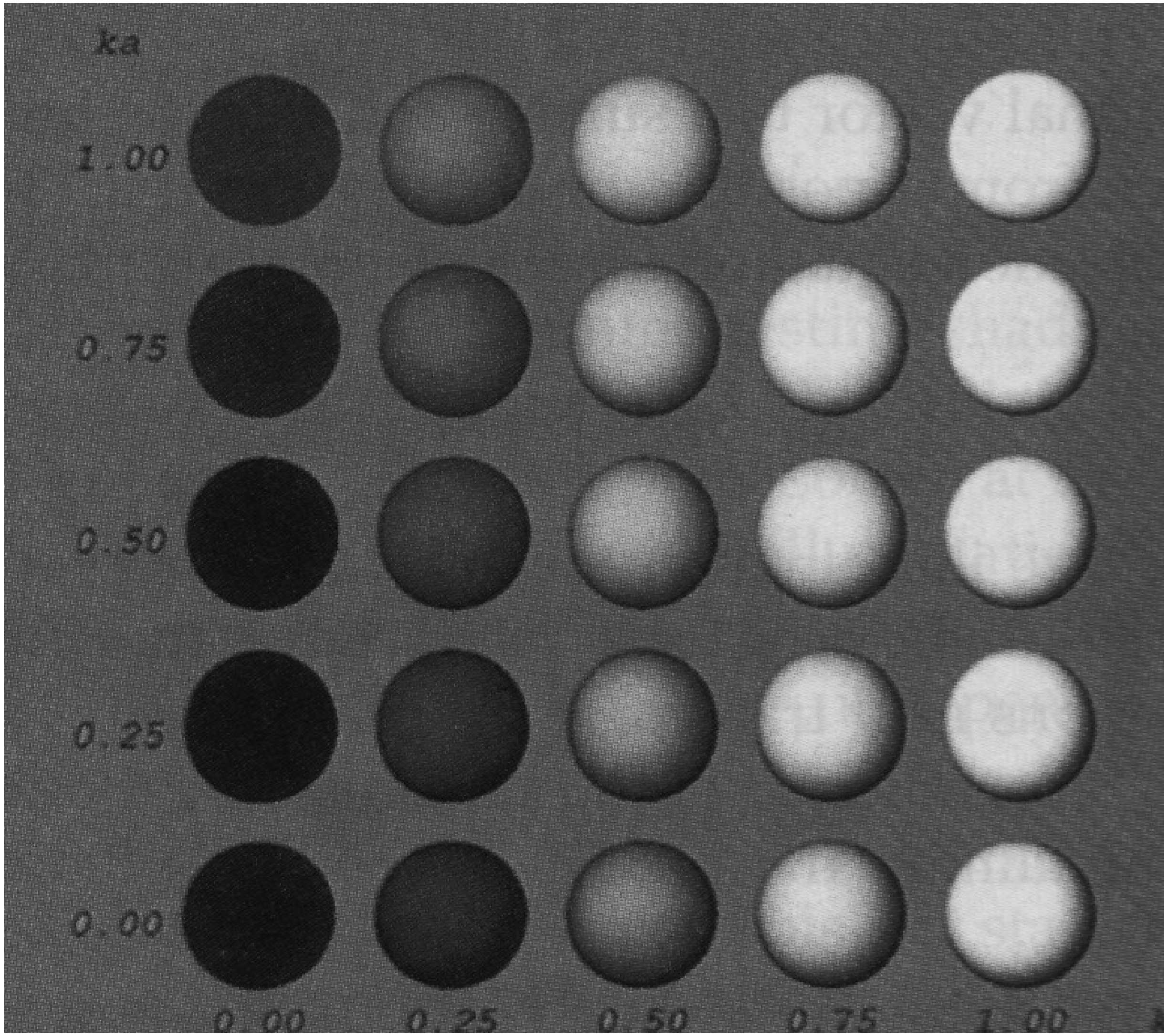
Basic Illumination Model

- Diffuse reflection for ambient light source + a point light source
  - $I_{diff} = k_a I_a + k_d I_l(\mathbf{N.L})$
- Fig:
  - sphere illuminated with different intensity ambient light
  - Illuminated with varying direction light source





Visual effects of different values of K_d and



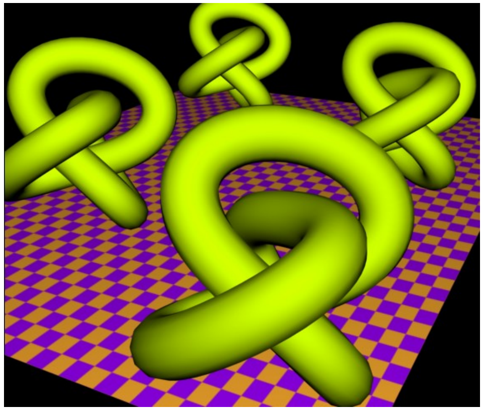
Diffuse Reflection - Further Discussions

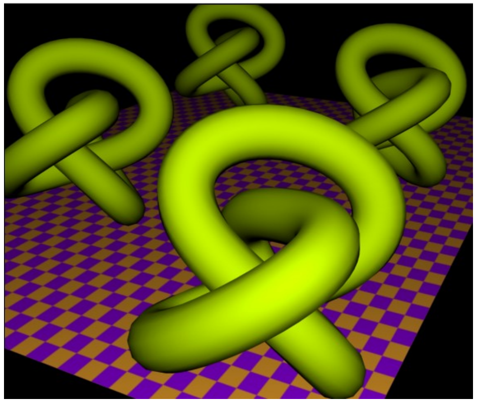
- Light-source attenuation

$$I = I_a k_a + f_{att} I_p k_d (N \cdot L)$$, where $f_{att} = \frac{1}{d_L^2}$

- Colored lights and surfaces
  - Similar for I_G and I_B.

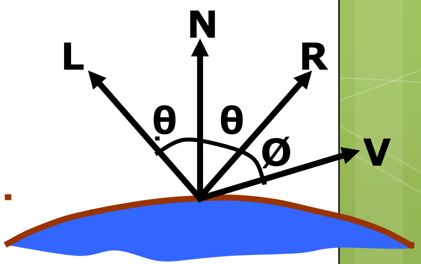
$$I_R = I_{aR} k_a O_{dR} + f_{att} I_{pR} k_d O_{dR} (N \cdot L)$$





Specular Reflection

- Bright spot seen at an illuminated shiny surface when viewed at certain direction
  - Polished metal surface, person's forehead, apple etc. exhibit specular reflection
- In fact an image of light source
- Result of total or near total reflection of incident light in a concentrated region around the specular reflection angle θ
- Fig:
  - **L** → unit vector pointing to light source
  - **N** → unit surface normal vector
  - **R** → unit vector in direction of specular reflection
  - **V** → unit vector pointing viewer
- Ideal reflector exhibit specelur reflection in the direction of R only (i.e Ø=0) but for non-ideal case specular reflection is seen over finite range of viewing positions



Phong Model

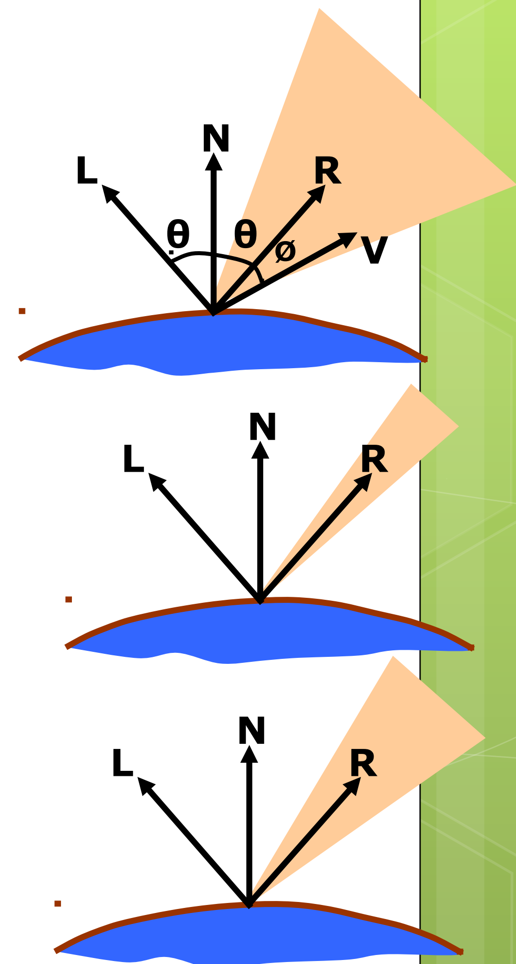
- Intensity of specular reflection: proportional to

$$\cos^{n_s} \phi$$

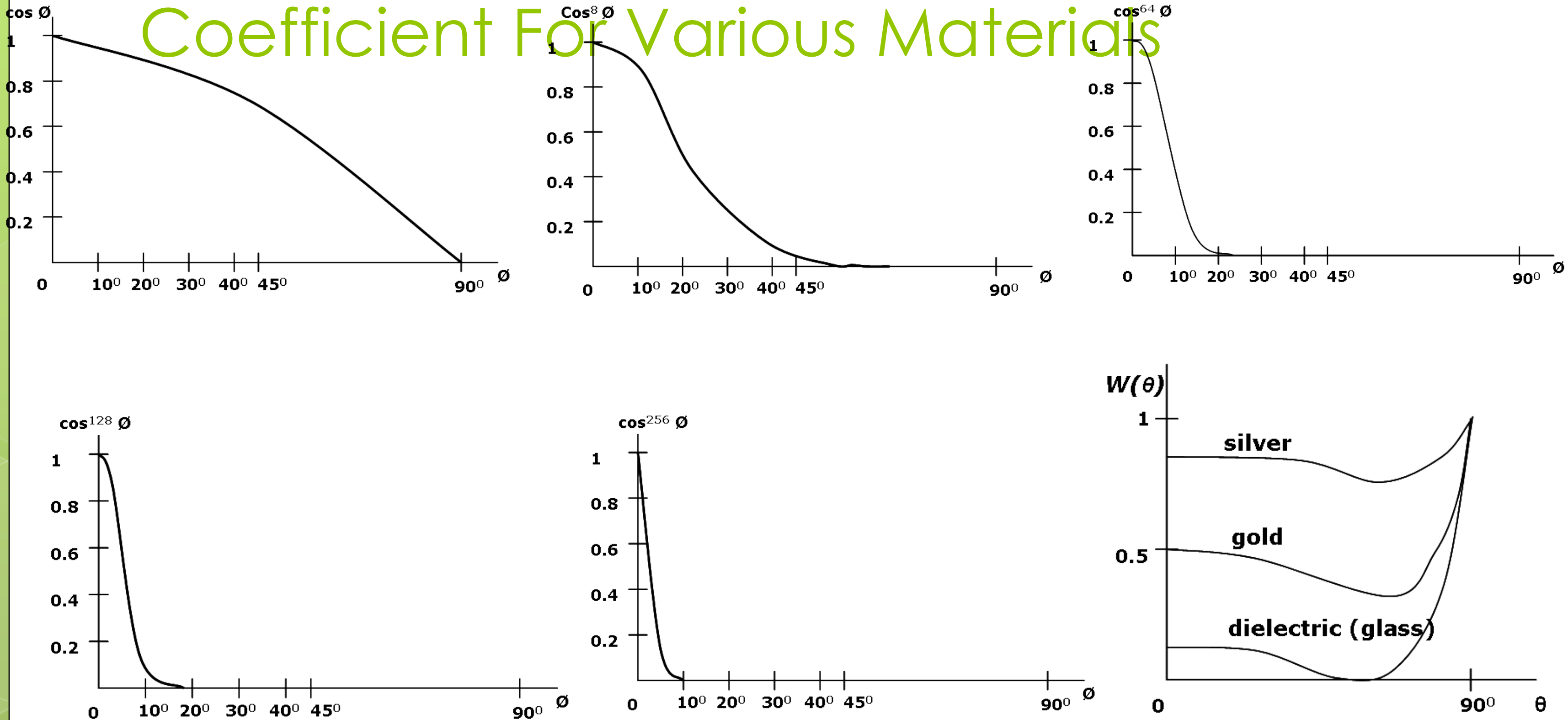
  - n_s → **specular reflection parameter** (depends on surface)
  - Ø ranges from 0 to 90^0 (i.e cos Ø varies from 0 to 1)
- Intensity of specular reflection depends on:
  - Material properties of surface
  - Angle of incidence θ
  - Other factors such as polarization and color of the incident light
- Monochromatic specular intensity variations can be approximated using **specular-reflection coefficient**, $w(\theta)$ for each surface

$$I_{spec} = w(\theta) I_l \cos^{n_s} \phi$$

- At $\theta = 90^0$, $w(\theta) = 1$ → all incident light is reflected



Plot For $\cos^{n_s} \phi$ and Specular Reflection Coefficient For Various Materials



Phong Model (contd...)

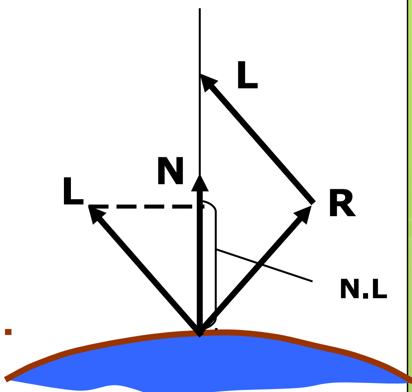
- Simplified form: assume $w(\theta) = k_s = constant$

$$I_{spec} = k_s I_l (V.R)^{n_s}$$

- Vector **R** can be evaluated from vectors **L** an **N** as:

  R + **L** = (2**N.L**)**N**

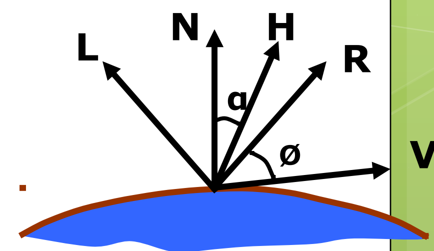
  i.e **R** = (2**N.L**)**N** – **L**



- Further simplified by replacing **V.R** with **N.H** where **H** is halfway vector between **L** and **V** (i.e **H** is unit bisector vector of angle between **L** and **V**)

$$H = \frac{L+V}{|L+V|}$$

- Thus $$I_{spec} = k_s I_l (N.H)^{n_s}$$



- If we add ambient light and diffuse reflection component then total intensity is given as:

$$\begin{aligned} I &= I_{diff} + I_{spec} \\ &= k_a I_a + k_d I_l (N.L) + k_s I_l (N.H)^{n_s} \end{aligned}$$

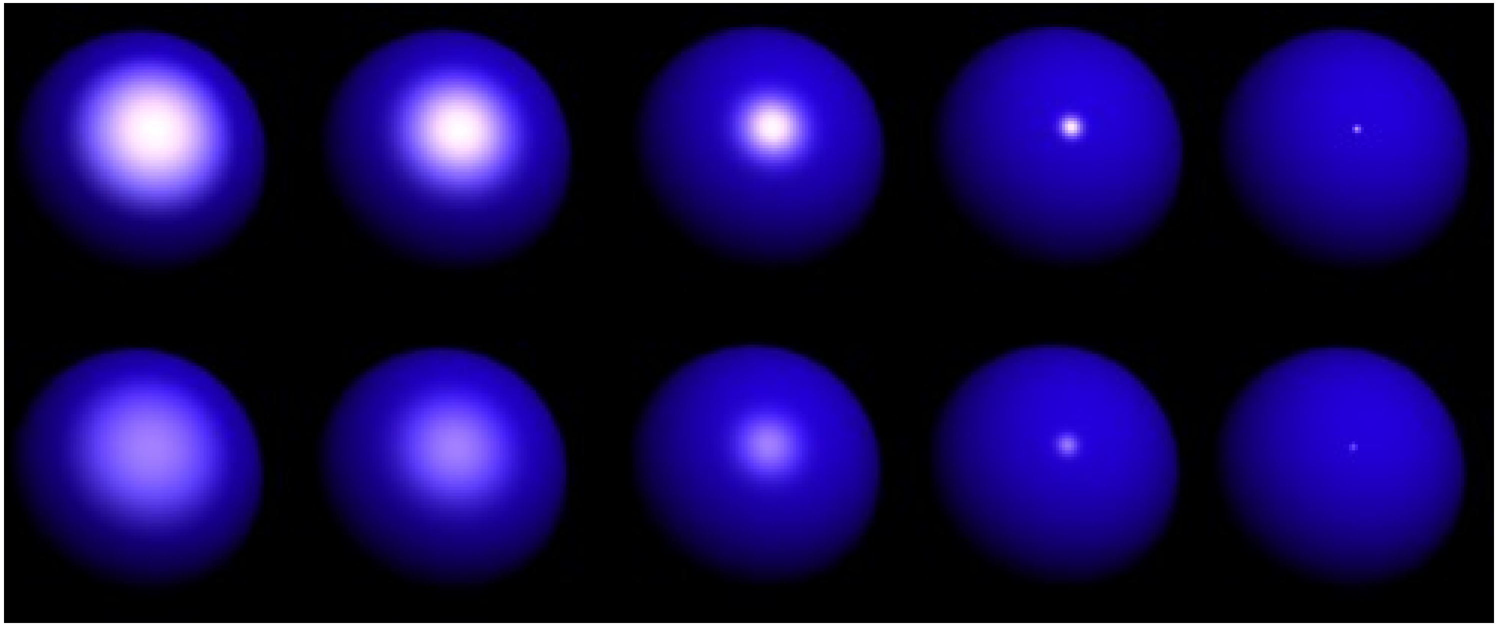
- For multiple light sources (n light sources)

$$I = k_a I_a + \sum_{i=1}^{n} I_{li} \left[k_d (N.L_i) + k_s (N.H_i)^{n_s} \right]$$

When v is coplanar with L and R
α = Ø/2 otherwise α > Ø/2

Visual Effects of Specular Reflections

K_s



1.0

0.5

8 16 32 64 128

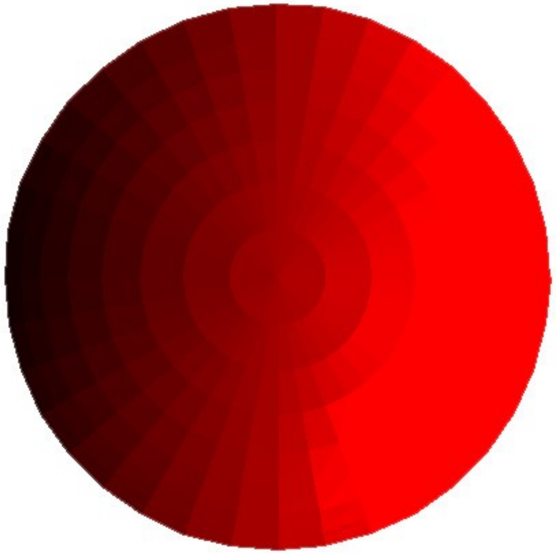
Shinness

Polygon Rendering Methods

- Illumination model is applied to fill the interior of polygons
- Curved surfaces are approximated with polygon meshes
  - But polyhedra that are not curved surfaces are also modeled with polygon meshes
- Two ways of polygon surface rendering
  - Single intensity for all points in a polygon
  - Interpolation of intensities for each point in a polygon
- Methods:
  - Constant Intensity Shading
  - Gouraud Shading
  - Phong Shading

Constant Intensity Shading

- Flat shading
  - Each polygon shaded with single intensity calculated for the polygon
- Useful for displaying general appearance of a curved surface
- Accurate rendering conditions:
  - Object is a polyhedron and not an curved surface approximation
  - All light sources should be sufficiently far from the surface (i.e **N.L** and attenuation function are constant over the polygon surfaces) :constant diffuse reflection??
  - Viewing position is sufficiently far (i.e **V.R** is constant over the surface) : Specular Reflection ??
  - ***Note****: Approximate rendering is possible even the conditions are not satisfied*
- Drawback: intensity discontinuity at the edges of polygons

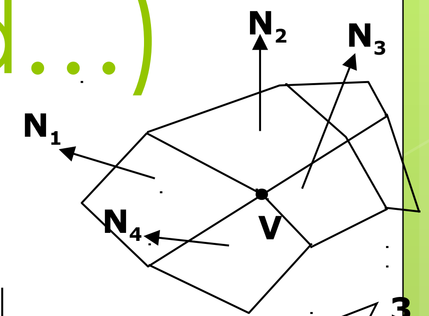


Credit goes to the students

Gouraud Shading

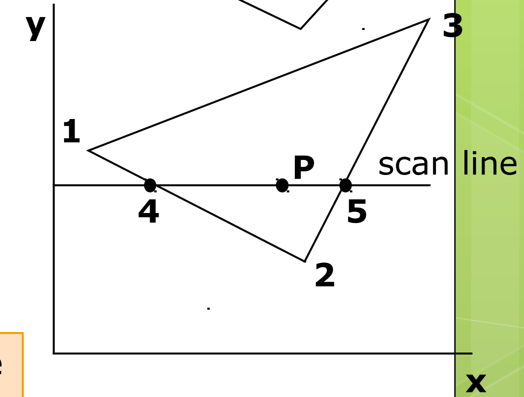
- Calculation Steps:
  - Determine the average unit normal vector at each polygon vertex
  - Calculate each of the vertex intensities by applying an illumination model
  - Linearly interpolate the vertex intensities over the polygon surface
- Intensity discontinuity at the edges of polygons is eliminated
- Drawback:
  - **Mach bands**: bright and dark intensity streaks caused by linear interpolation of intensities
    - Could be reduced by dividing the surface into large number of polygons or by using other methods, such as *Phong shading*

Gouraud Shading (contd…)



- Average Unit Normal: Obtained by averaging the surface normals of all polygons sharing the vertex

$$N_v = \frac{\sum_{k=1}^{n} N_k}{\left|\sum_{k=1}^{n} N_k\right|}$$

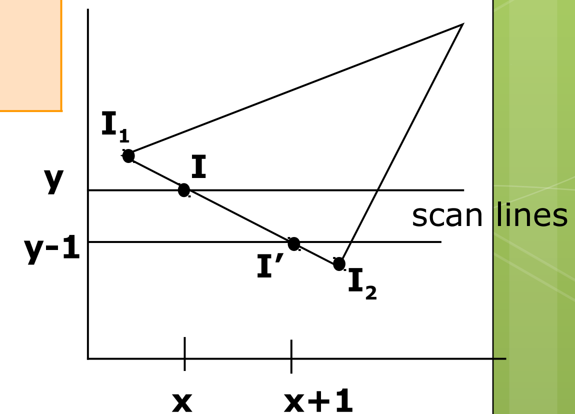


- Intensity interpolation:
  - Along the polygon edges are obtained by interpolating intensities at the edge ends

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

Recursive calculation along the edge

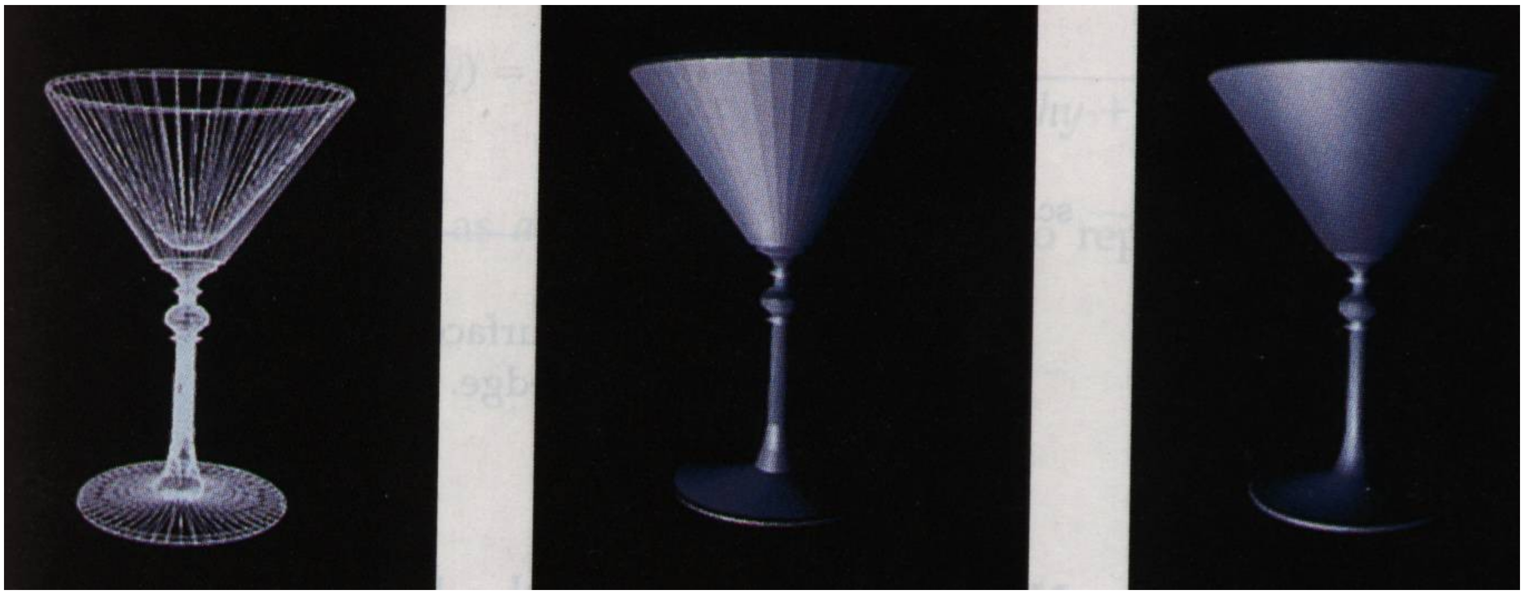
$$I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$



  - Along the scan line between the polygon edges are obtained by interpolating intensities at the intersection of scan line and polygon edges

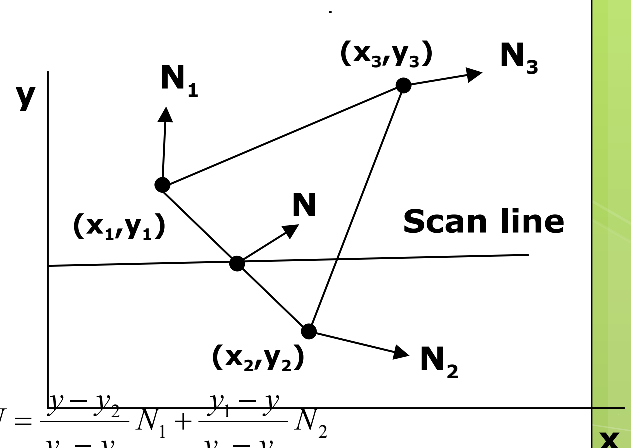
$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - y_4} I_5$$

Recursive Calculation along the scan line ??



Phong Shading

- More accurate method for rendering
- Fundamental: *Interpolate normal vectors and apply illumination model to each surface point*
- Calculation steps:
  - Determine average unit normal vectors at each polygon vertex
  - Linearly interpolate the vertex normals over the surface of the polygon
  - Apply an illumination model along each scan line to calculate projected pixel intensities for the surface points
- **Trade-off**: requires considerably more calculations

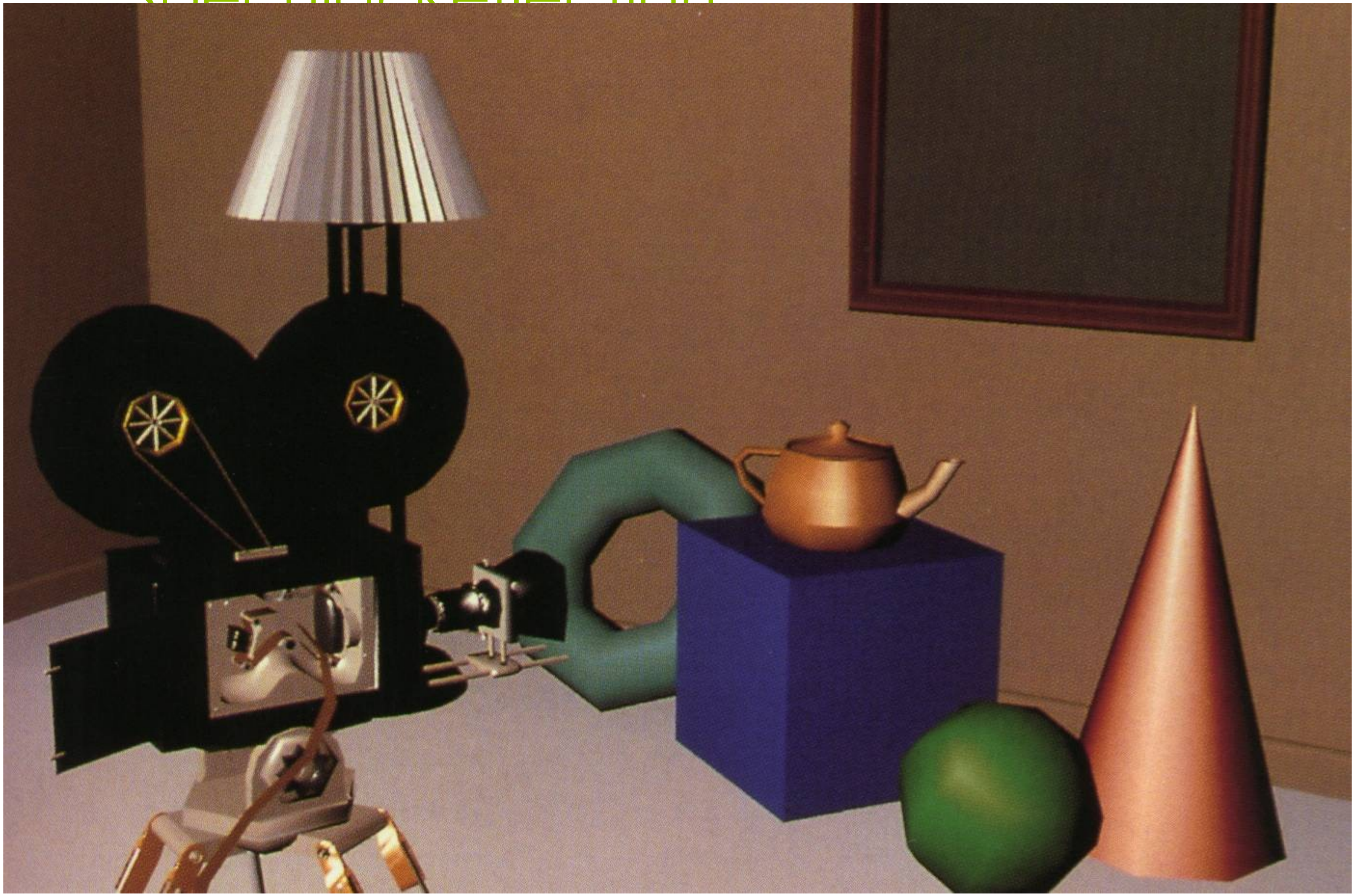


$$N = \frac{y - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y}{y_1 - y_2} N_2$$

Note: Students are encouraged to read Fast Phong Shading which could be useful for project works

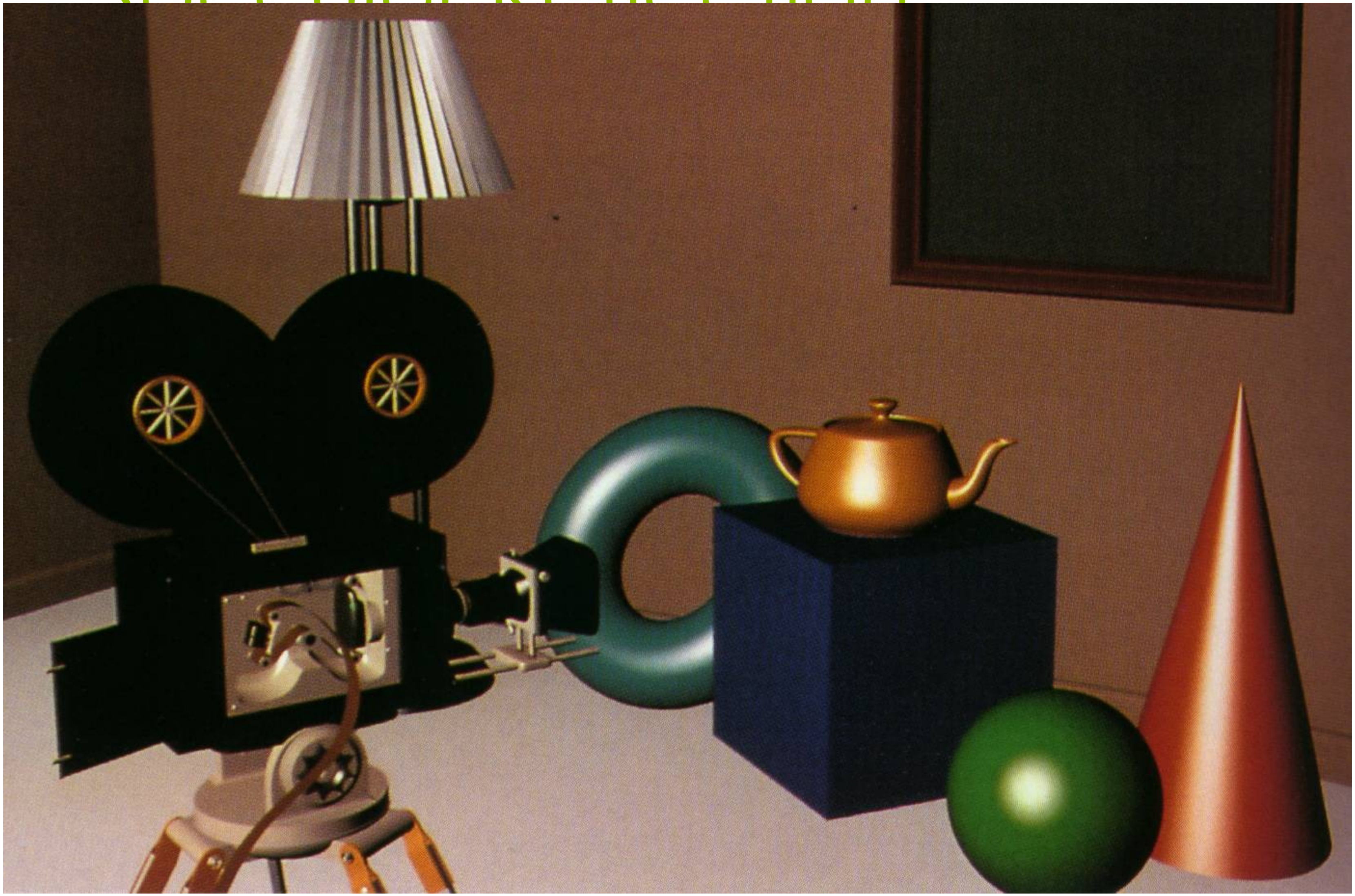
Gouraud Shaded Polygons with Diffuse Reflection

Gouraud Shaded Polygons with Diffuse and Specular Reflection



Phong Shaded Polygons

Curved Surfaces with Specular Reflection



The Rendering Pipeline

- For Z-buffer and Phong shading