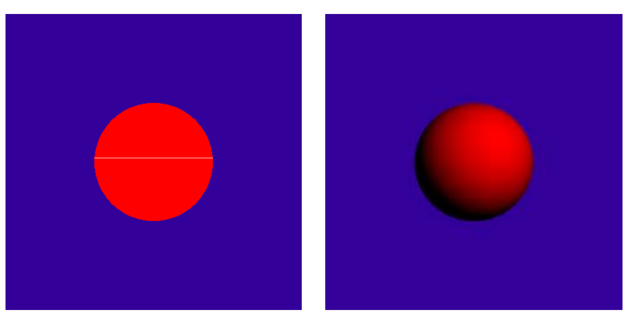
# CS 362: Computer Graphics

**Illumination**

Dr. Samit Bhattacharya
Dept. of Comp. Sc. & Engg.,
IIT Guwahati, Assam, India

## Why Lighting?

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



## Lighting

- To create realistic image, we need to simulate the *lighting* of the surfaces in the scene
  - Fundamentally simulation of *physics* and *optics*
  - A lot of approximations are done to do this simulation fast enough

## Definitions

- **Illumination**: the transport of energy from light source to surfaces & points
  - Note: includes *direct* and *indirect illumination*
- **Lighting**: the process of computing the luminous intensity (i.e., outgoing light) at a particular 3-D point, usually on a surface
- **Shading/surface rendering**: the process of assigning colors to pixels (why the distinction?)

## Definitions

- Illumination models :
  - **Empirical**: simple formulations that approximate observed phenomenon
  - **Physically based**: models based on the actual physics of light interacting with matter
- Empirical models are mostly used in interactive graphics for simplicity
- Increasingly, realistic graphics are using physically based models

## Components of Illumination

- Light source and surface properties
- Light source (or *emitters*)
  - Spectrum of emittance (i.e., color of the light)
  - Geometric attributes
    - Position
    - Direction
    - Shape
  - Directional attenuation
  - Polarization

## Components of Illumination

- Surface properties
  - Reflectance spectrum (i.e., color of the surface)
  - Subsurface reflectance
  - Geometric attributes
    - Position
    - Orientation
    - Micro-structure
- Simplifications done in CG
  - Only *direct illumination* from emitters to surfaces
  - Simplify geometry of emitters to trivial cases

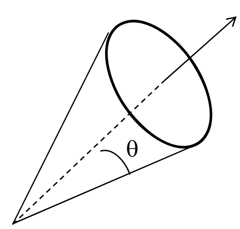
## Ambient Light Source

- Objects not directly lit are typically still visible
  - Result of *indirect illumination* from emitters, bouncing off intermediate surfaces
- Too expensive to calculate (in real time), so a simplification is used, called an *ambient light source*
  - No spatial or directional characteristics; illuminates all surfaces equally
  - Amount reflected depends on surface properties

## Point Light Source

- A *point light source* emits light equally in all directions from a single point
  - The direction to the light from a point on a surface differs for different points
- Special case - an *infinitely distant light source*
  - Direction is constant for all surfaces in the scene
  - All rays of light from the source are parallel
    - A good approximation to sunlight
- **The direction from a surface to the light source is important in lighting the surface**

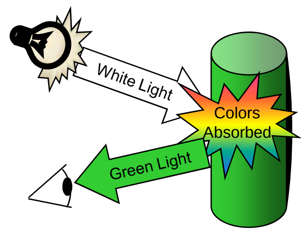
## Directional Light Source (spotlight)




- Required to produce a “beam of light” effect
- Can be set up using
  - Position
  - Color
  - A vector direction (spot light direction)
  - An angular limit θ
- Object outside angular limit, no illumination by the light source

# Reflected Light

- The colors we perceive are determined by the nature of the light reflected from an object
  - e.g. when white light is shone onto a green object, most wavelengths are absorbed, while green light is reflected from the object




# Surface Lighting Effects

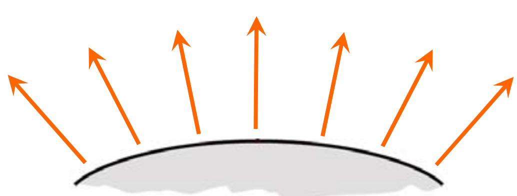
- The amount of incident light reflected by a surface depends on the type of material
  - Shiny materials reflect more of the incident light
  - Dull surfaces absorb more of the incident light
- For transparent surfaces, some of the light is also transmitted through the material

## Surface Lighting Effects

- Reflections
  - Diffuse
  - Specular

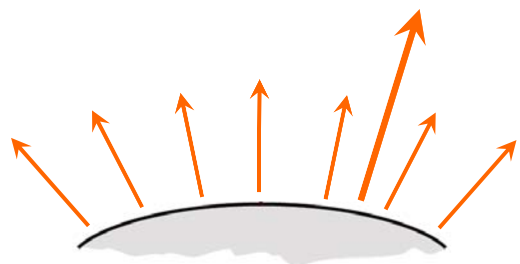
## Diffuse Reflection

- Surfaces that are rough or grainy tend to reflect light in all directions
  - This scattered light is called **diffuse reflection**



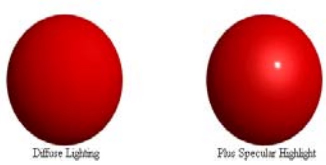
## Specular Reflection

- Some of the reflected light is concentrated into a highlight or bright spot
  - This is called **specular reflection**



## Specular Reflection

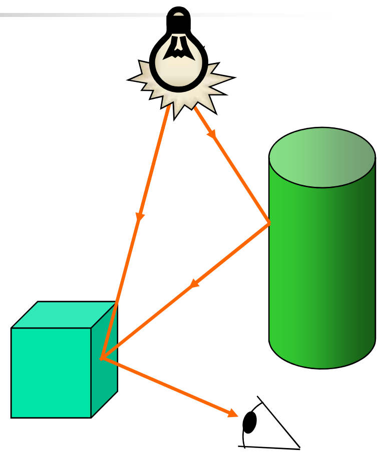
- Shiny surfaces exhibit *specular reflection*
  - Polished metal
  - Glossy car finish



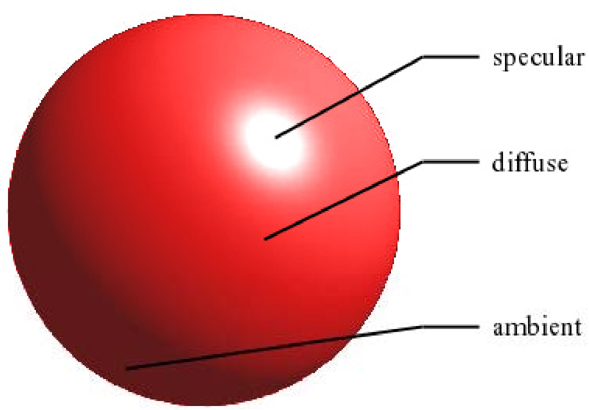

- A light shining on a specular surface causes a bright spot known as a *specular highlight*
  - Where these highlights appear is a function of the viewer's position, so **specular reflectance is view dependent**

## Ambient Light

- A surface that is not exposed to direct light may still be lit up by reflections from other nearby objects – **ambient light**
- The total reflected light from a surface is the sum of the contributions from light sources and ambient light



## Example



## Basic Illumination Model

- A simplified model of illumination
  - Gives reasonably good results
  - Used in most graphics systems
- The important components are
  - Ambient light
  - Diffuse reflection
  - Specular reflection
- Assume monochromatic light, initially

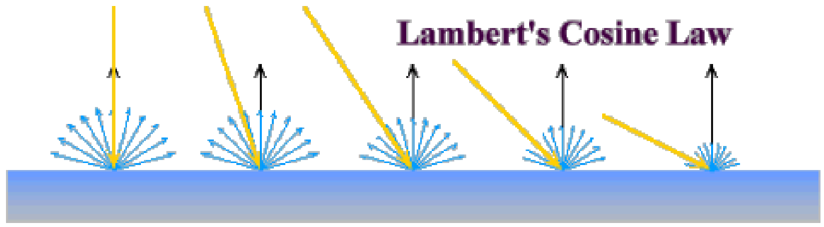
## Ambient Light

- To incorporate ambient (background) light, simply set a general brightness level for a scene
  - Denote this value as $I_a$
- This approximates the global diffuse reflections from various surfaces within the scene

## Diffuse Reflection

- Assumption-surfaces reflect incident light with equal intensity in all directions
  - Such surfaces are called **ideal diffuse reflectors** or **Lambertian reflectors -** Follows Lambert's Cosine Law
    - The energy reflected by a small portion of a surface from a light source in a given direction is proportional to the cosine of the angle between that direction and the surface normal
- Note: the reflected intensity is independent of the viewing direction, but depends on the surface orientation with regard to the light source

## Lambert's Law



## Diffuse Reflection Contd…

- A parameter $k_d$ is set for each surface
  - Determines the fraction of incident light that is to be scattered as diffuse reflections from that surface (the **diffuse-reflection coefficient** or the **diffuse reflectivity**)
- $k_d$ is assigned a value between 0.0 and 1.0
  - 0.0: dull surface that absorbs almost all light
  - 1.0: shiny surface that reflects almost all light

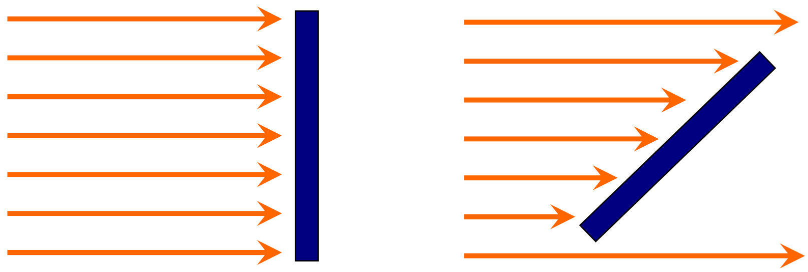
## Diffuse Reflection – Ambient Light

- For background lighting effects, assume that every surface is fully illuminated by the scene's ambient light $I_a$
- Therefore, the ambient contribution to the diffuse reflection is

$$I_{ambdiff} = k_d I_a$$

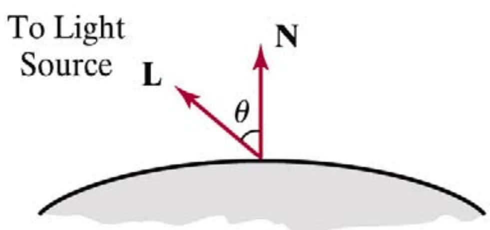
## Diffuse Reflection

- When a surface is illuminated by a light source, the amount of incident light depends on the orientation of the surface relative to the light source direction



## Diffuse Reflection

- The angle between the incoming light direction and a surface normal is referred to as the **angle of incidence** given as $\theta$



## Diffuse Reflection

- The amount of incident light on a surface can be approximated as (following Lambert's law)

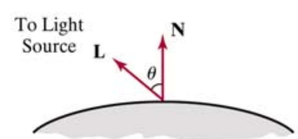
$$I_{l,incident} = I_l \cos\theta$$

- Thus, the diffuse reflections component is

$$\begin{aligned} I_{l,diff} &= k_d I_{l,incident} \\ &= k_d I_l \cos\theta \end{aligned}$$

## Diffuse Reflection

- Let $N$ denotes surface normal, $L$ denotes unit direction vector to the light source
- Then, $N \cdot L = \cos\theta$




- Thus,

$$I_{l,diff} = \begin{cases} k_d I_l (N \cdot L) & \text{if } N \cdot L > 0 \\ 0 & \text{if } N \cdot L \le 0 \end{cases}$$

## Combining Ambient and Incident Diffuse Reflections

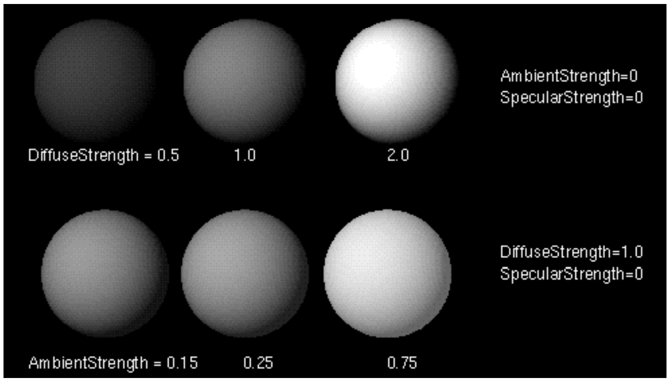
- To combine the diffuse reflections arising from ambient and incident light, most graphics packages use two separate diffuse-reflection coefficients:
  - $k_a$ for ambient light
  - $k_d$ for incident light

## Combining Ambient and Incident Diffuse Reflections

- The total diffuse reflection equation for a single point source can then be given as:

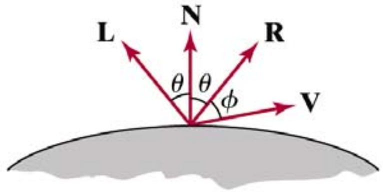
$$I_{diff} = \begin{cases} k_a I_a + k_d I_l (N \cdot L) & \text{if } N \cdot L > 0 \\ k_a I_a & \text{if } N \cdot L \le 0 \end{cases}$$

## Example



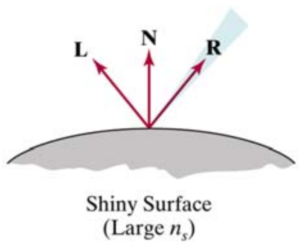

## Specular Reflection

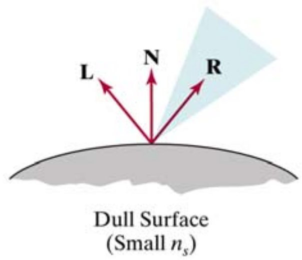
- The bright spot on a shiny surface
  - Result of near total reflection of the incident light in a concentrated region around the **specular reflection angle**
  - The specular reflection angle equals the angle of the incident light (Snell's law)



## Specular Reflection

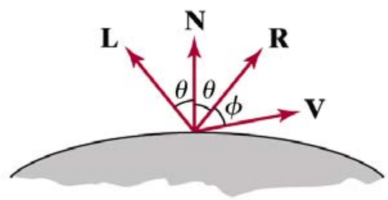
- A perfect mirror reflects light only in the specular-reflection direction
- Other objects exhibit specular reflections over a finite range of viewing positions around vector *R*




## The Phong Specular Reflection Model

- An empirical model for calculating specular reflection range - Phong Bui Tuong (1973)
- The intensity of specular reflection as proportional to the angle between the viewing vector and the specular reflection vector

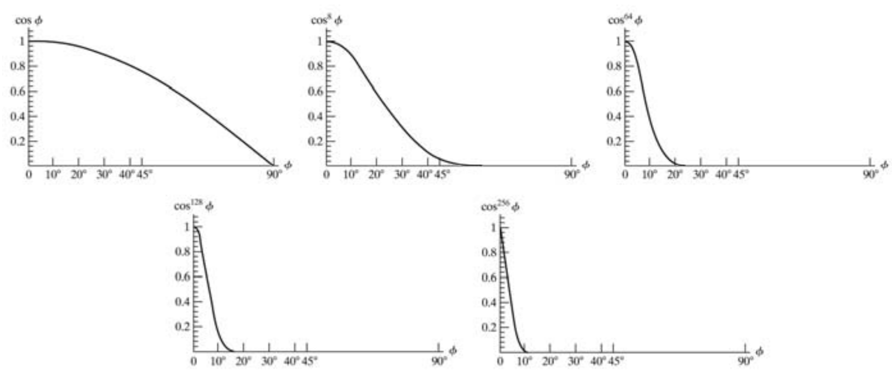


# The Phong Specular Reflection Model

- The specular reflection intensity is proportional to $\cos^{n_s} \phi$
- The angle $\Phi$ can be varied between 0° and 90° so that $\cos\Phi$ varies from 1.0 to 0.0
- The specular reflection exponent, $n_s$ is determined by the type of surface we want to display
  - Shiny surfaces have a very large value (>100)
  - Rough surfaces would have a value near 1

# The Phong Specular Reflection Model

- The effect of $n_s$ on the angular range in which specular reflections can be seen



# The Phong Specular Reflection Model

- For some materials the amount of specular reflection depends on the angle of the incident light
  - *Fresnel's Laws of Reflection* describe how specular reflections behave
- However, it is possible to approximate the specular effects (for most opaque objects) with
  - A constant specular reflection coefficient $k_s$

# The Phong Specular Reflection Model

- The specular reflection intensity

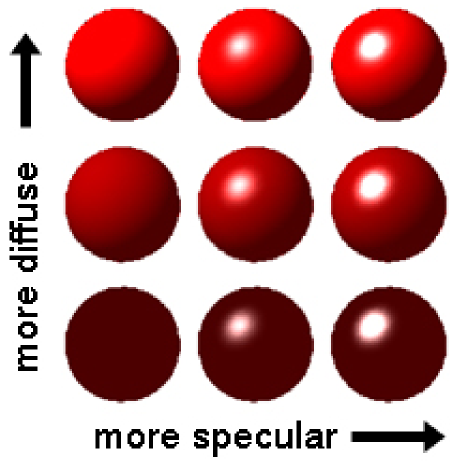
$$I_{l,spec} = k_s I_l \cos^{n_s} \phi$$

- Note that $V \cdot R = \cos\phi$ and no specular reflections if $V$ and $L$ are on the same side of $N$

$$I_{l,spec} = \begin{cases} k_s I_l (V \cdot R)^{n_s} & \text{if } V \cdot R > 0 \text{ and } N \cdot L > 0 \\ 0.0 & \text{if } V \cdot R < 0 \text{ or } N \cdot L \leq 0 \end{cases}$$

- $R$ can be represented as $(2N.L)N\text{-}L$

# Example




# Combining Reflections

- For a single light source, the effects of diffuse and specular reflections can be combined as follows

$$I = I_{diff} + I_{spec}$$
$$= k_a I_a + k_d I_l (N \cdot L) + k_s I_l (V \cdot R)^{n_s}$$

## Multiple Light Sources

- A scene may contain any number of light sources
- The diffuse and specular reflections computed as sums of the contributions from the various sources

$$I = I_{ambdiff} + \sum_{l=1}^{n} \left[ I_{l,diff} + I_{l,spec} \right]$$

$$= k_a I_a + \sum_{l=1}^{n} I_l \left[ k_d \left( N \cdot L \right) + k_s \left( V \cdot R \right)^{n_s} \right]$$

## Intensity Attenuation

- As light moves from a light source its intensity diminished
- If this fact is not considered, then all surfaces illuminated with equal intensities (assuming same optical characteristics), irrespective of their distance from light source
  - May result in indistinguishable overlapping of two surfaces when projected on screen

# Radial Intensity Attenuation

- As light moves from a light source its intensity diminished
  - At any distance $d_l$ away from the light source the intensity diminishes by a factor of $1/d_l^2$
- However, using the factor $1/d_l^2$ does not produce very good results
  - For small d, too much intensity variation
  - For large d, too small intensity variations
- The above arise because real scene doesn't have point light source, but our model is too simple to account for real light effect

# Radial Intensity Attenuation

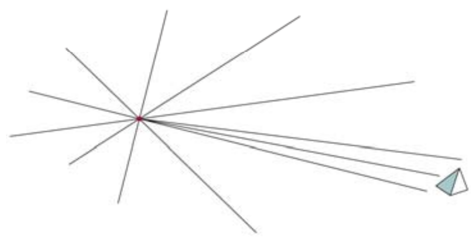
- An effective and useful approximation - an inverse quadratic function

$$f_{radatten}(d_l) = \frac{1}{a_0 + a_1 d_l + a_2 d_l^2}$$

the coefficients $a_0$, $a_1$, and $a_2$ can be varied to produce optimal results

## Infinitely Distant Light Sources

- A large light source, like the sun, can be modelled as a point light source
  - However, it will have very little directional effect
- Radial intensity attenuation is not used (value=1)



## Angular Intensity Attenuation

- Light intensity also decreases angularly – primarily used for spotlights
- A commonly used function for calculating angular attenuation is:

$$f_{angatten}(\phi) = \begin{cases} 1.0 & \text{if light source not spotlight} \\ 0.0 & \text{if object outside spotlight} \\ \cos^{a_l}\phi & \text{otherwise} \end{cases}$$

the attenuation exponent $a_l$ is assigned some positive value and angle $\phi$ is measured from the spotlight cone axis

## Adding Intensity Attenuation

- To incorporate radial and angular intensity attenuation, we simply adjust our equation to take these into account
- So, light intensity is now given as

$$I = I_{ambdiff} + \sum_{l=1}^{n} \left[ f_{l,radatten} f_{l,angatten} \left( I_{l,diff} + I_{l,spec} \right) \right]$$

where $f_{radatten}$ and $f_{angatten}$ are the corresponding attenuations

## RGB Colour Considerations

- For an RGB colour description each intensity specification is a three element vector
- So, for each light source $I_l = \left( I_{lR}, I_{lG}, I_{lB} \right)$
- Similarly all parameters are given as vectors

$$k_a = \left( k_{aR}, k_{aG}, k_{aB} \right) \quad k_d = \left( k_{dR}, k_{dG}, k_{dB} \right)$$

$$k_s = \left( k_{sR}, k_{sG}, k_{sB} \right)$$

## RGB Colour Considerations

- Each component of the surface colour is then calculated with a separate expression
- For example

$$I_{lR,diff} = k_{dR} I_{lR} (N \cdot L)$$

$$I_{lG,diff} = k_{dG} I_{lG} (N \cdot L)$$

$$I_{lB,diff} = k_{dB} I_{lB} (N \cdot L)$$

## Lighting Review

- Lighting Model
  - Ambient
    - Normals don't matter
  - Lambert/Diffuse
    - Angle between surface normal and light
  - Phong/Specular
    - Surface normal, light, and viewpoint

## Applying Illumination

- We now have an illumination model for a point on a surface
- Assuming that our surface is defined as a mesh of polygonal facets, *which points should we use?*

## Applying Illumination

- Keep in mind:
  - It's a fairly expensive calculation
  - Several possible answers, each with different implications for the visual quality of the result
- Shading/surface rendering methods

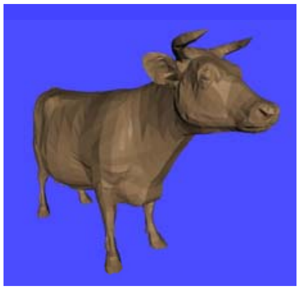
## Flat Surface Rendering

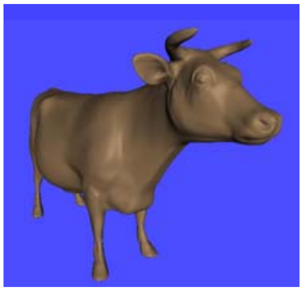
- The simplest method for rendering a polygon surface
- The same color is assigned to all surface positions
- The illumination at a single point on the surface is calculated and used for the entire surface
- Flat surface rendering is extremely fast, but can be unrealistic

## Flat Surface Rendering

- The method works well if the following conditions are met
  - The object is a polyhedron and is not an approximation of an object with curved surfaces
  - All light source should be sufficiently far from the surface so that **N.L** and the attenuation are constant over the surface
  - Viewing position is sufficiently far from the surface so that **V.R** is constant over the surface

## Overcoming Flat Shading Limitations





- Just add lots and lots of polygons – however, this is SLOW!

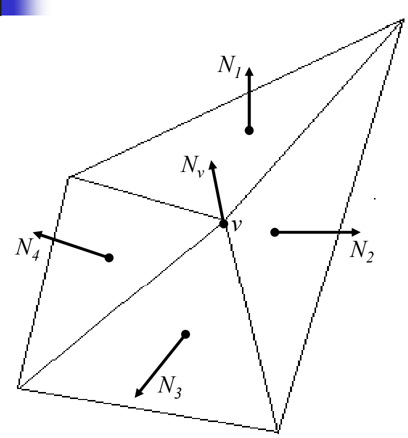
## Gouraud Surface Rendering

- Developed in the 1970s by Henri Gouraud
  - Also called **intensity-interpolation surface rendering**
- Intensity levels are calculated at each vertex and interpolated across the surface

# Gouraud Surface Rendering

- Steps
  - Determine the average unit normal vector at each vertex of the polygon
  - Apply an illumination model at each polygon vertex to obtain the light intensity at that position
  - Linearly interpolate the vertex intensities over the projected area of the polygon

# Gouraud Surface Rendering




- The average unit normal vector at $v$ is given as

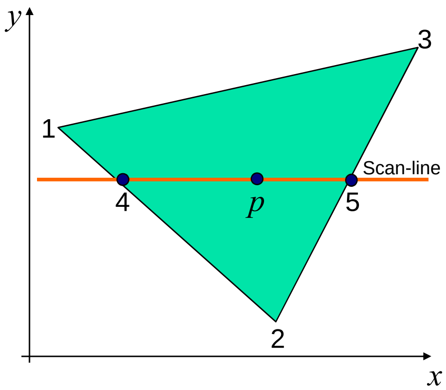
$$N_v = \frac{N_1 + N_2 + N_3 + N_4}{|N_1 + N_2 + N_3 + N_4|}$$

- or more generally

$$N_v = \frac{\sum_{i=1}^{n} N_i}{\left|\sum_{i=1}^{n} N_i\right|}$$

## Gouraud Surface Rendering

- Illumination values are linearly interpolated across each scan-line

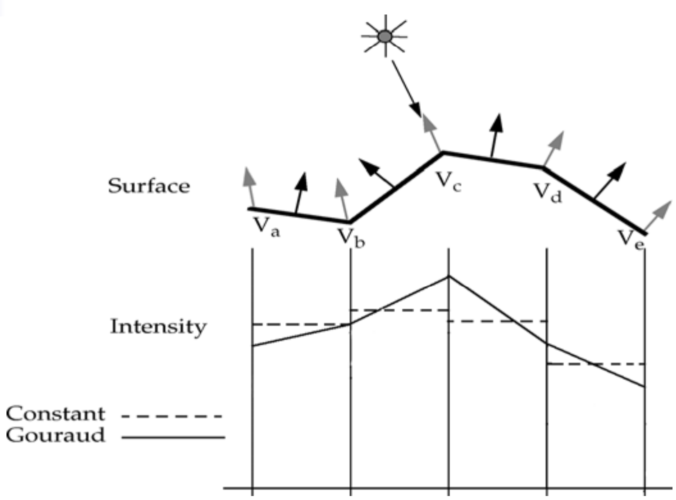



$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

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

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

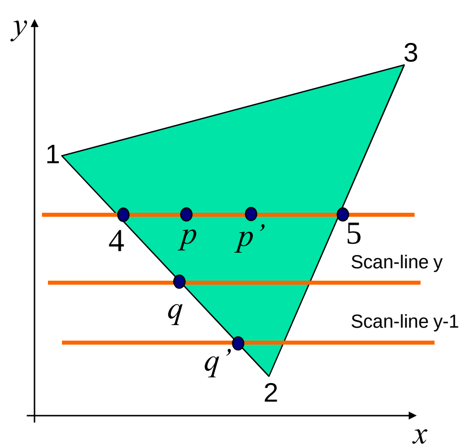
## Advantages



# Implementation

- Gouraud surfacing rendering can be implemented relatively efficiently using an iterative approach
- Typically Grouaud shading is implemented as part of a visible surface detection technique

# Gouraud Surface Rendering (Iterative Method)

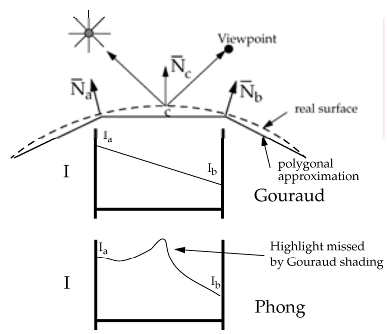



$$I_{p'} = I_p + \frac{I_5 - I_4}{x_5 - x_4}$$

$$I_{q'} = I_q + \frac{I_2 - I_1}{y_1 - y_2}$$

## Problems With Gouraud Shading

- Tends to miss certain highlighting. In particular, problem with specular reflections
- Can introduce anomalies known as **Mach bands**
  - A psychological phenomenon whereby we see bright bands where two blocks of solid colour meet



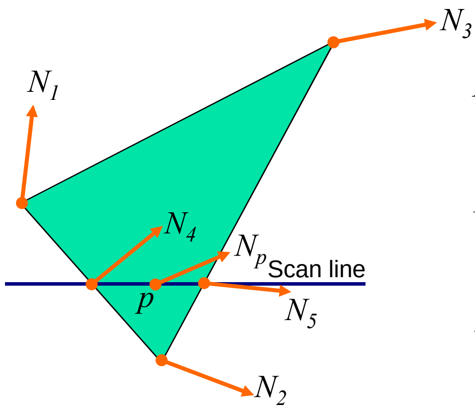

## Phong Surface Rendering

- A more accurate interpolation based approach for rendering a polygon
- The Phong surface rendering model (or **normal-vector interpolation rendering**) interpolates normal vectors instead of intensity values

# Phong Surface Rendering

- To render a polygon, Phong surface rendering proceeds as follows
  - Determine the average unit normal vector at each vertex of the polygon
  - Linearly interpolate the vertex normals over the projected area of the polygon
  - Apply an illumination model at positions along scan lines to calculate pixel intensities using the interpolated normal vectors


# Phong Surface Rendering


$$N_4 = \frac{y_4 - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y_4}{y_1 - y_2} N_2$$

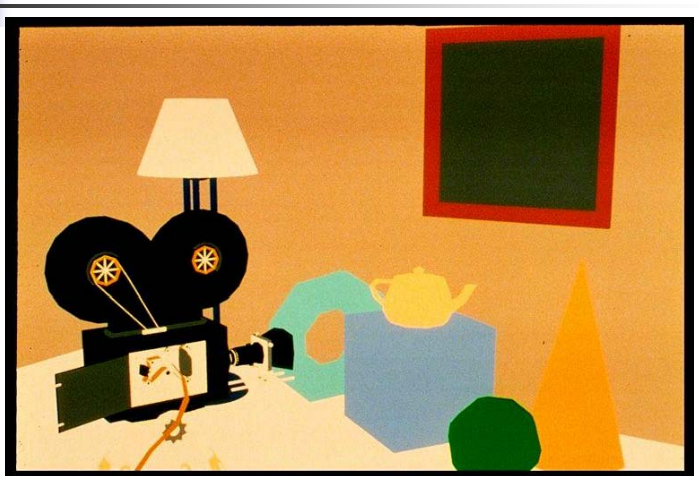
$$N_5 = \frac{y_5 - y_2}{y_3 - y_2} N_3 + \frac{y_3 - y_5}{y_3 - y_2} N_2$$

$$N_p = \frac{x_5 - x_p}{x_5 - x_4} N_4 + \frac{x_p - x_4}{x_5 - x_4} N_5$$

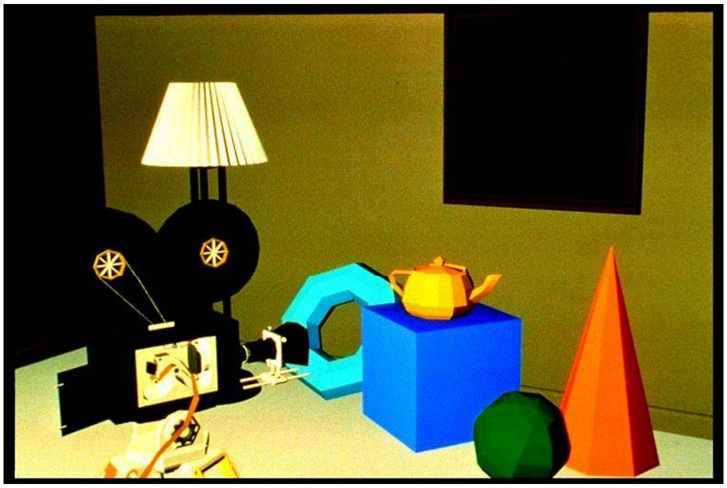
## Implementation

- Phong shading is much slower than Gouraud shading as the lighting model is revaluated so many times
- However, there are fast Phong surface rendering approaches (H&B, pp 595-596)
- Typically Phong shading is implemented as part of a visible surface detection technique

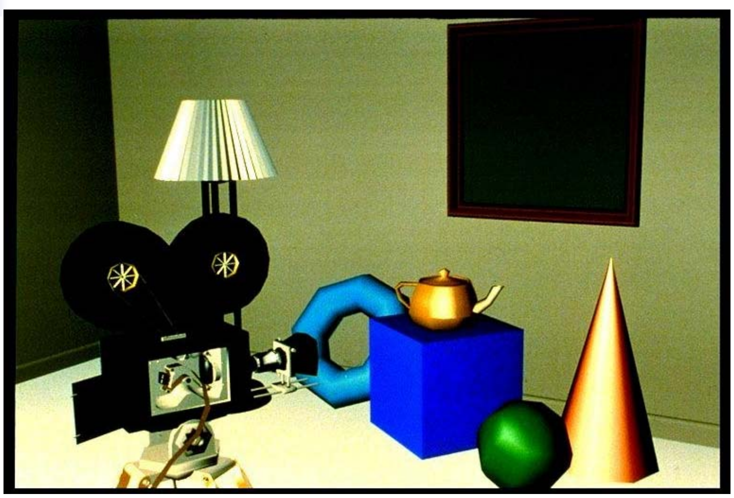
## No Surface Rendering



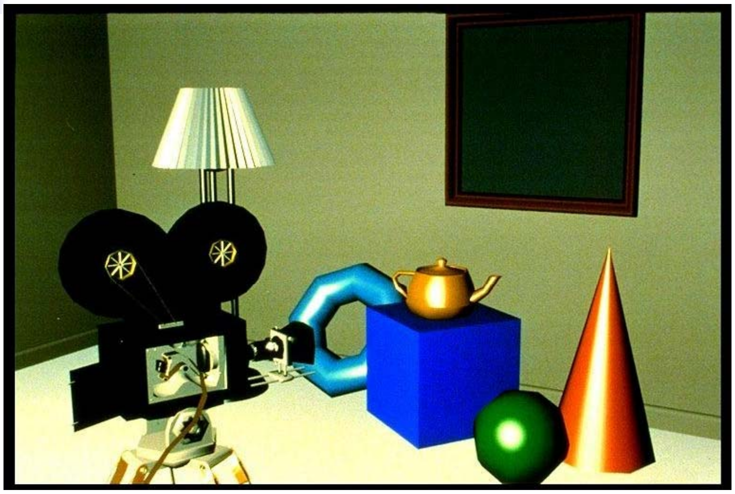
## Flat Surface Rendering



## Gouraud Surface Rendering



## Phong Surface Rendering



## Summary

- For realistic rendering of polygons we need interpolation methods to determine lighting positions
- Flat shading is fast, but unrealistic
- Gouraud shading is better, but does not handle specular reflections very well
- Phong shading is better still, but can be slow