Lighting/Illumination/Reflection Model
Model

BRAC UNIVERSITY
Inspiring Excellence

Some of the contents used in this lecture belong to…

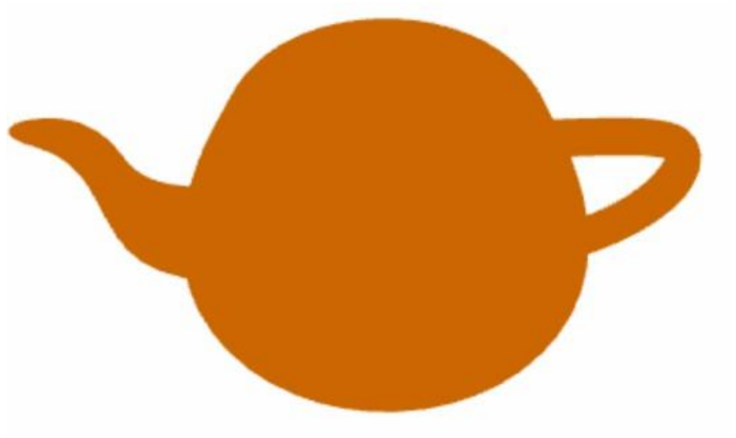
- Dr Jon Shiach, Manchester Metropoliton University, “Direct Lighting Model”, https://www.youtube.com/watch?v=7CdS8oOJtVA
- Mr. Jacobson, University of Toronto, “WebGL Phong Shading”, http://www.cs.toronto.edu/~jacobson/phong-demo/
- RapidCompact, “Simplifying a 3D Mesh”, https://rapidcompact.com/doc/cli/latest/Simplify/index.html

BRAC UNIVERSITY
Inspiring Excellence

Basic Terms

- Illumination: the transport of energy from light sources to surfaces & points
  - Local illumination
  - Global illumination
- Lighting model or Illumination model: Express the factors determining a surface's color or luminous intensity (outgoing or reflected light) at a particular 3D point

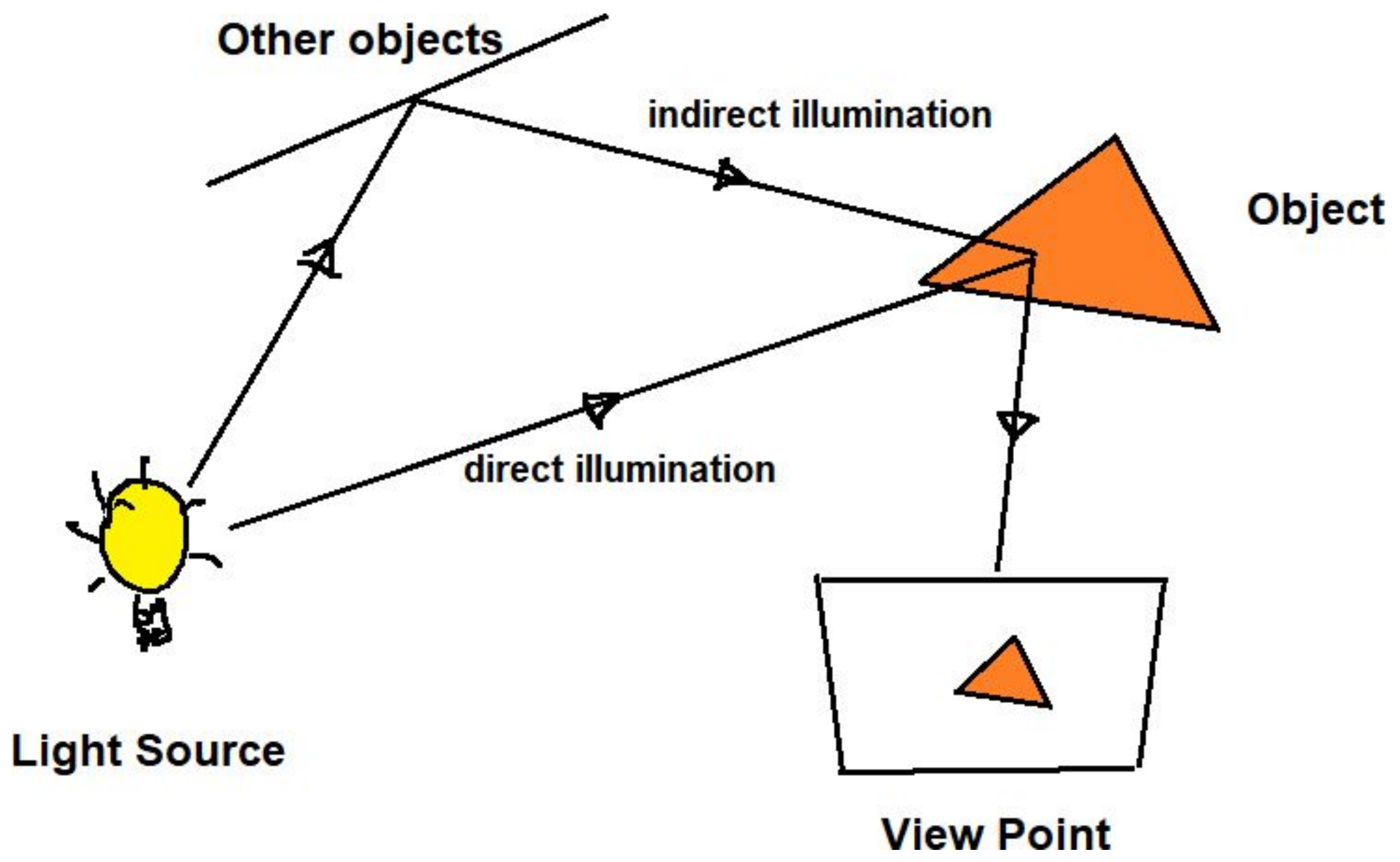
Effects of Lighting



Pixel colours only

Pixel colours and lighting

Illumination

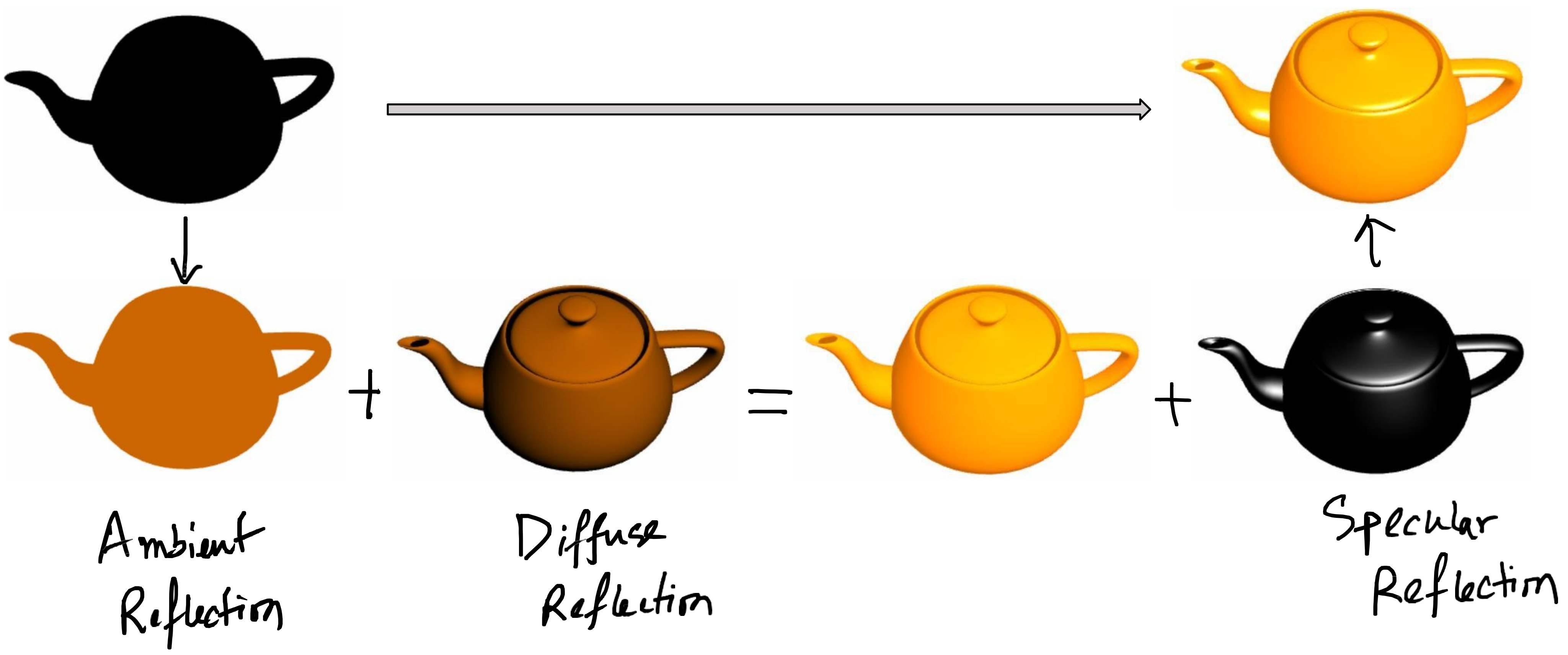


Components of Illumination

- Two components of illumination:
  1. Light sources
  2. Surface properties
- Light source described by a luminance/intensity ‘I’
  - Each color is described separately
  - I = [Ir Ig Ib]
- Types of Light Sources:
  1. Ambient Light
  2. Diffuse Light
  3. Spot Light

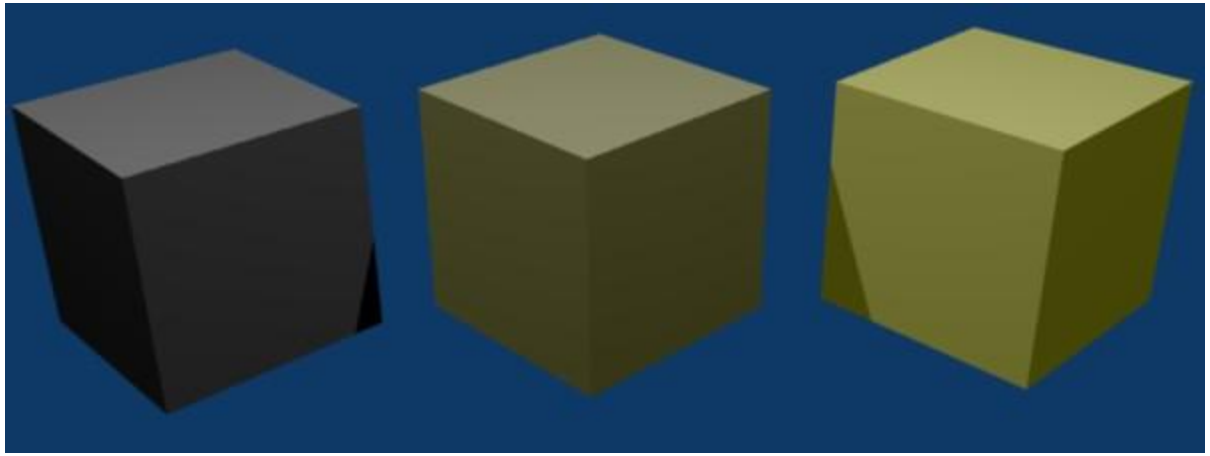
BRAC UNIVERSITY
Inspiring Excellence

Phong's Reflection Model



Ambient Light

- No identifiable source or direction
- Product of multiple reflections of light from the many surfaces present in the environment
- Computationally inexpensive



Ambient Light

<u>Categories:</u>

1. Global ambient light
   - Independent of light source
   - Lights entire scene
   - Example: reflection of sunlight from several surfaces
2. Local ambient light
   - Contributed by additional light sources
   - Can be different for each light and primary color
   - Example: Reflection of fluorescent lamps from several surfaces

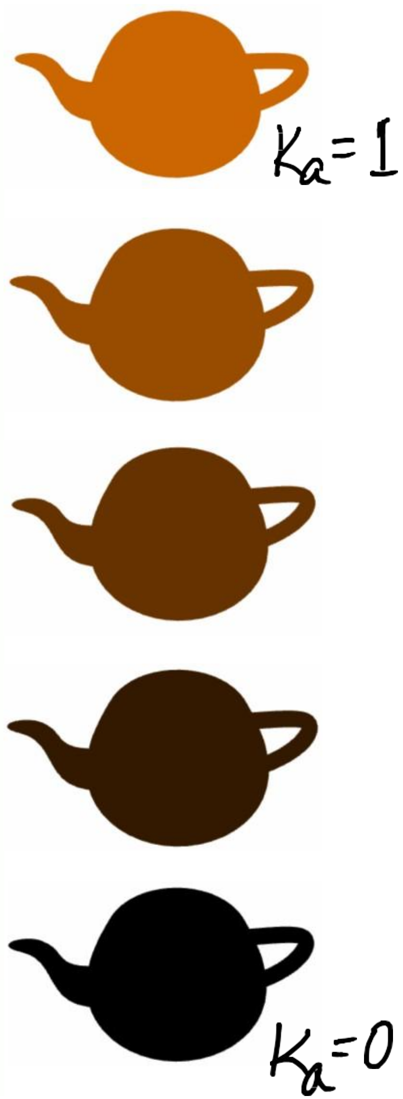
Ambient Reflection

- **Ambient reflection** is the reflection of light that does not come directly from a light source
- Even in a darkened room, we can make out the edges of objects – this is because of light bouncing off of objects
- Since Phong is a **direct** lighting model, we assume that ambient light falls equally on all objects, i.e.,

$$A = I_a k_a$$

where I_a is the intensity of the ambient light and $k_a \in [0, 1]$ is the **ambient coefficient**

- k_a is set to provide the right amount of ambient light for a scene, e.g., $k \to 1$ for bright scenes and $k \to 0$ for dark or nighttime scenes.



Ambient Reflection Coefficient

- Effect of adding ambient light to the diffuse light reflected by a sphere
- Diffuse source intensity is 1.0
  Diffuse reflection coefficient is 0.4
  Ambient source intensity is 1.0
- Moving from left to right the ambient reflection coefficient takes on values: 0.0, 0.1, 0.3, 0.5, and 0.7
  - Too little ambient light makes shadows too deep and harsh
  - Too much makes the picture look washed out and bland

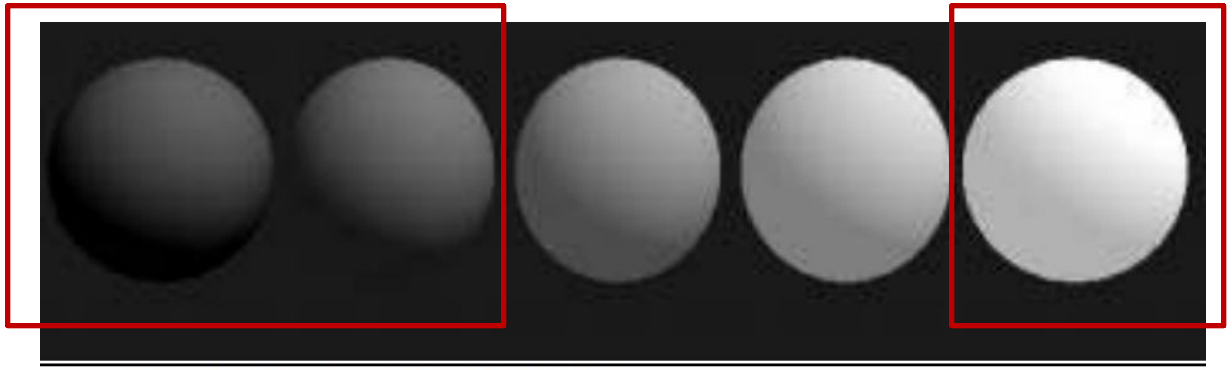
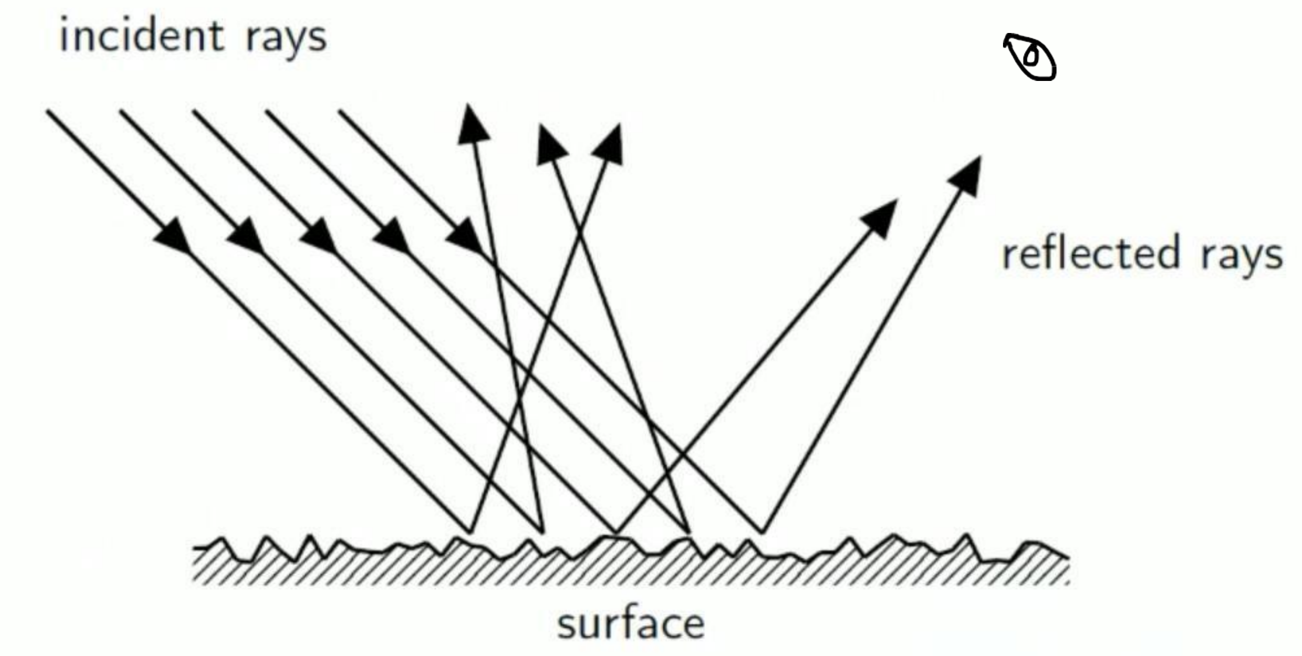


Figure 8.16. On the effect of ambient light.

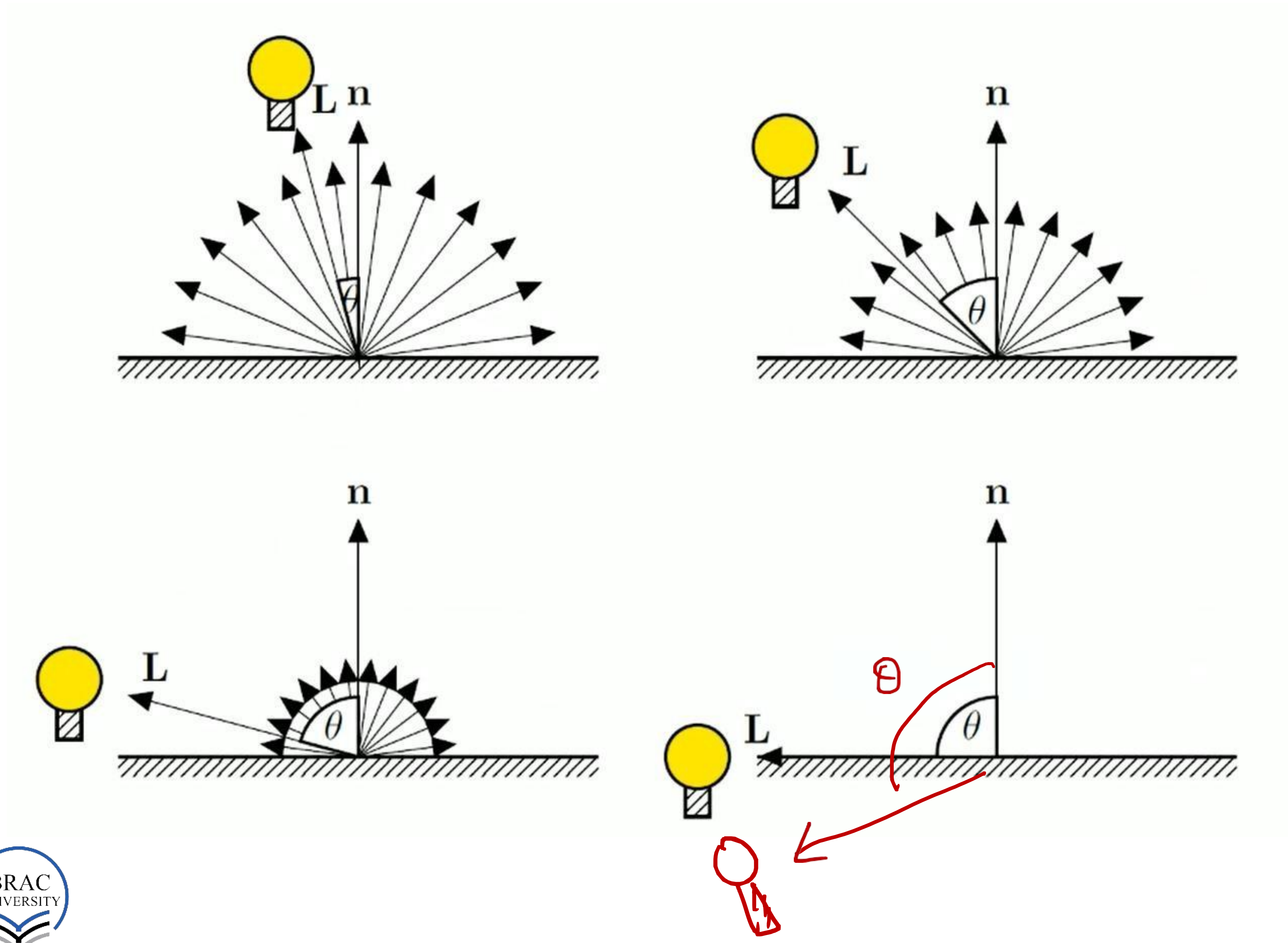
BRAC UNIVERSITY
Inspiring Excellence

Diffuse Reflection

- Parallel rays from light source
- Reflected rays are scattered
- Direction of reflection
- Some are visible, some are not



Diffuse Reflection – Phong's model



- Point light source
- Light reflected equally in all direction
- "Magnitude" of reflection depends on θ
- Direction of L reversed? → (−L)

Diffuse Reflection – Phong's model

- Phong's diffuse reflection model depends upon the position of the light source relative to the surface

$$D = I_p k_d \max[\underline{\cos(\theta)}, 0]$$

where

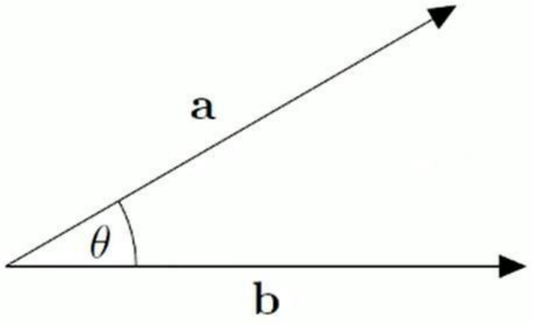
  - I_p is the intensity of the point light source
  - $k_d \in [0, 1]$ is the **diffuse coefficient**
  - θ is the angle between the lighting vector and the surface normal

- The $\max[\cos(\theta), 0]$ is used so that no light is reflected if the light source is behind the surface

BRAC UNIVERSITY
Inspiring Excellence

Diffuse Reflection – Phong's model

Dot product $\longleftrightarrow$ cos(θ)



- The definition of the dot product is

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}| \cos(\theta)$$

$a.b = \cos(\theta)$

- If $\mathbf{L}$ and $\mathbf{n}$ are unit vectors then

$$\mathbf{L} \cdot \mathbf{n} = \cos(\theta)$$

so we can replace the expensive cosine calculation by an easy dot product, i.e,

$$D = I_p k_d \max(\mathbf{L} \cdot \mathbf{n}, 0)$$

BRAC UNIVERSITY
Inspiring Excellence

Diffuse Reflection Coefficient

- $I_d = \max\{I_s k_d \cos\theta, 0\}$
- Source intensity is 1.0
- Background intensity is 0.4
- Sphere reflecting diffuse light, for six reflection coefficients: 0, 0.2, 0.4, 0.6, 0.8, and 1.

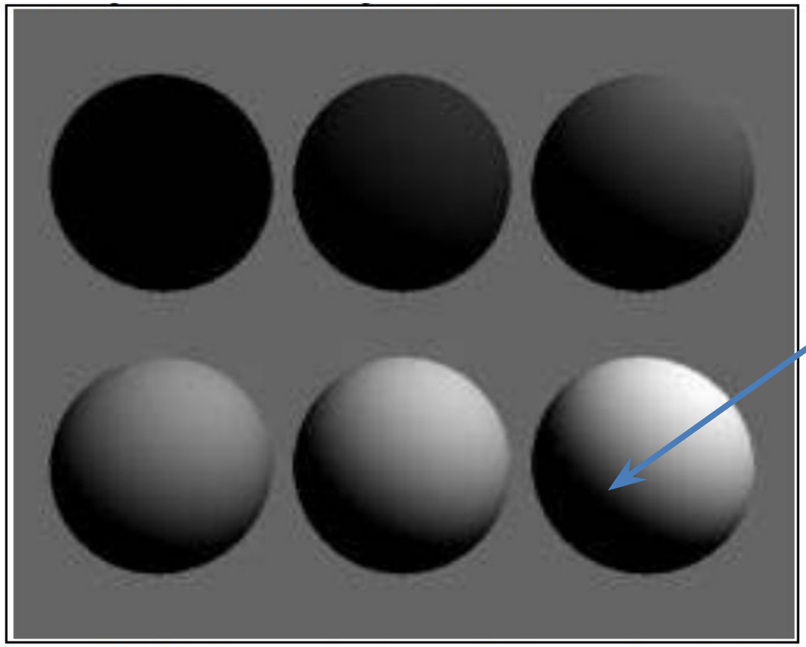


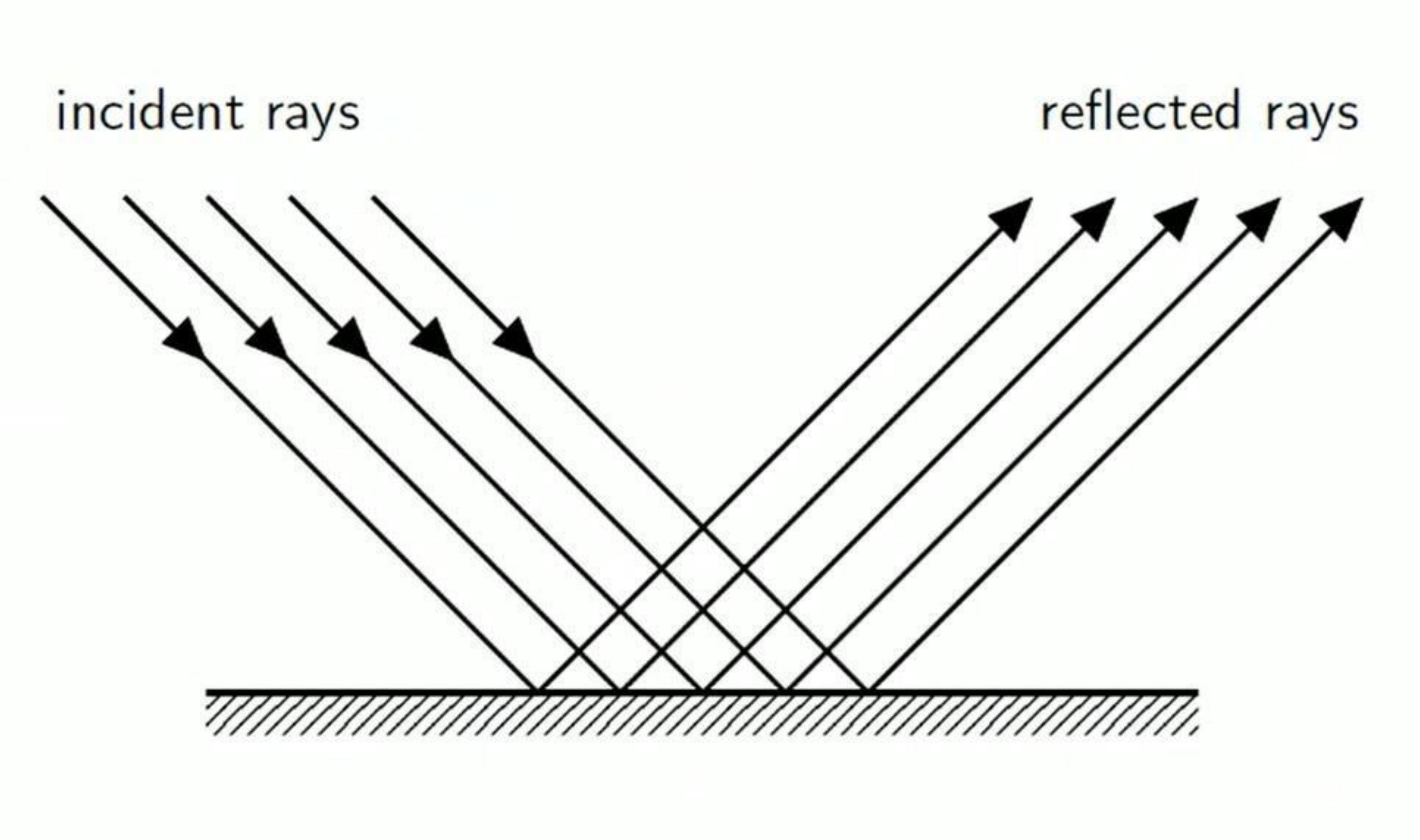
Figure 8.11. Spheres with various reflection coefficients shaded with diffuse light. (file: fig8.11.bmp)

Angle θ between surface normal and incident light is > 90º

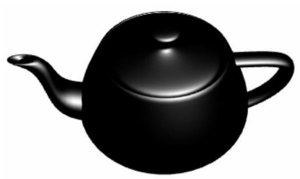
What is the ambient component here?

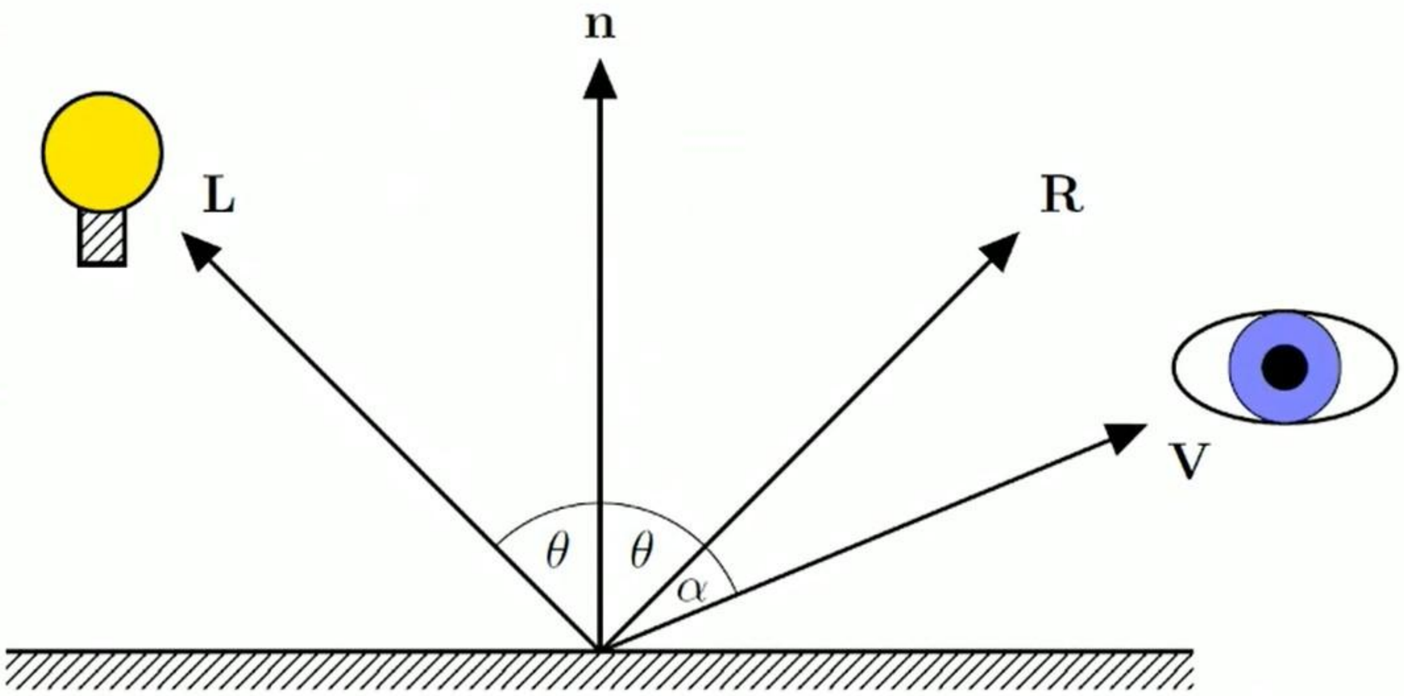
What is the specular component?

Specular Reflection



Specular Reflection





Specular Reflection

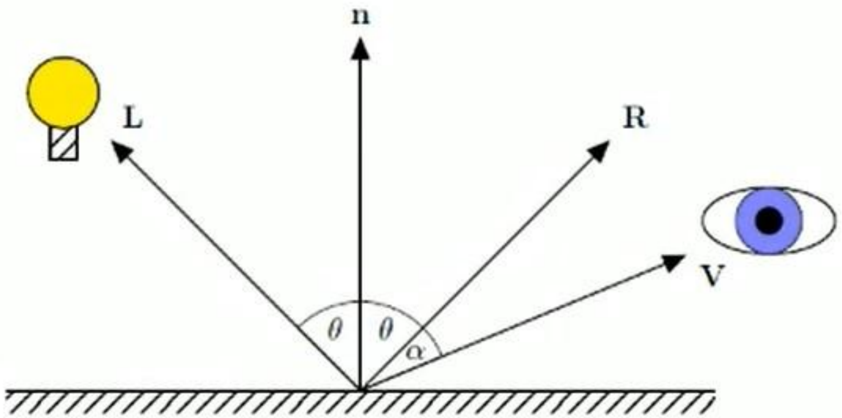
- Phong's specular model is

$$S = I_p k_s \cos^n(\alpha)$$

where

  - $k_s \in [0, 1]$ is the **specular coefficient**
  - n is the **specular exponent** (shininess)
  - α is the angle between $\mathbf{R}$ and $\mathbf{V}$

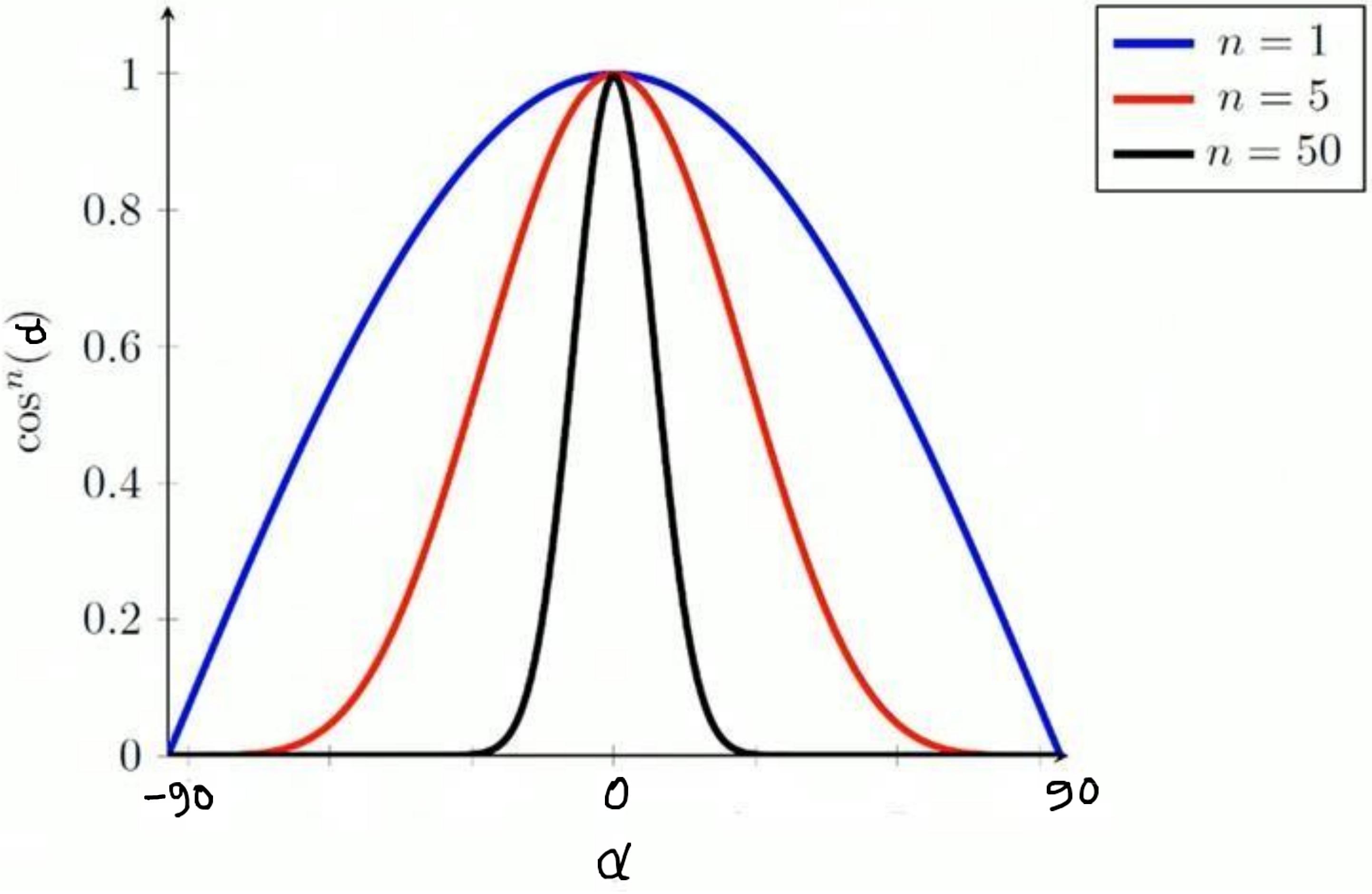
- The $\cos^n(\alpha)$ term determines the amount of light that is reflected



Specular Reflection

Shininess

$(0.2)^5$



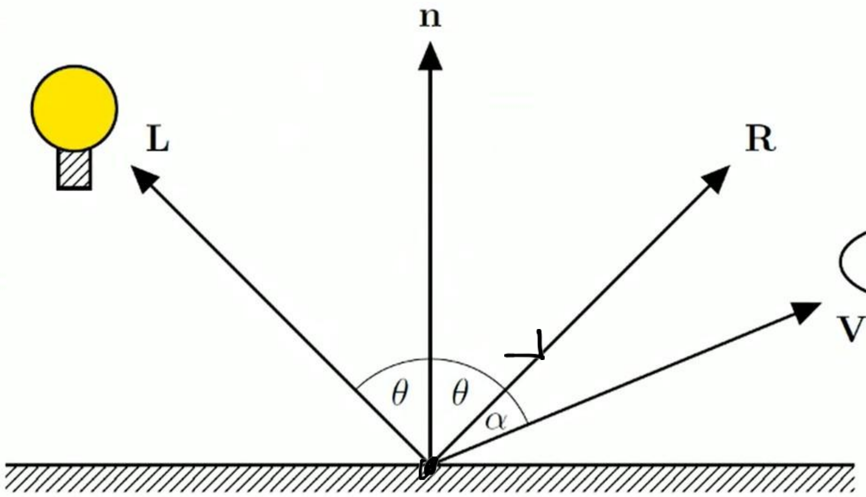
Specular Reflection

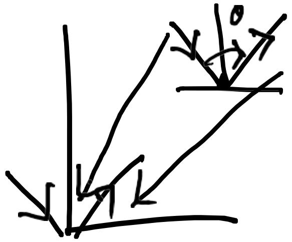
$$S = I_p k_s \cos^n(\alpha)$$

Calculate α, $\cos(\alpha)$

- $\cos(\alpha) = \underline{V.R}$ [unit vectors]

- R
  - Using Rotation
  - Projection of vector





Specular Reflection Coefficient

- The ambient and diffuse reflection coefficients are 0.1 and 0.4 for all spheres.
- $I_d = I_s\, k_s\, (\cos \varphi)^k$

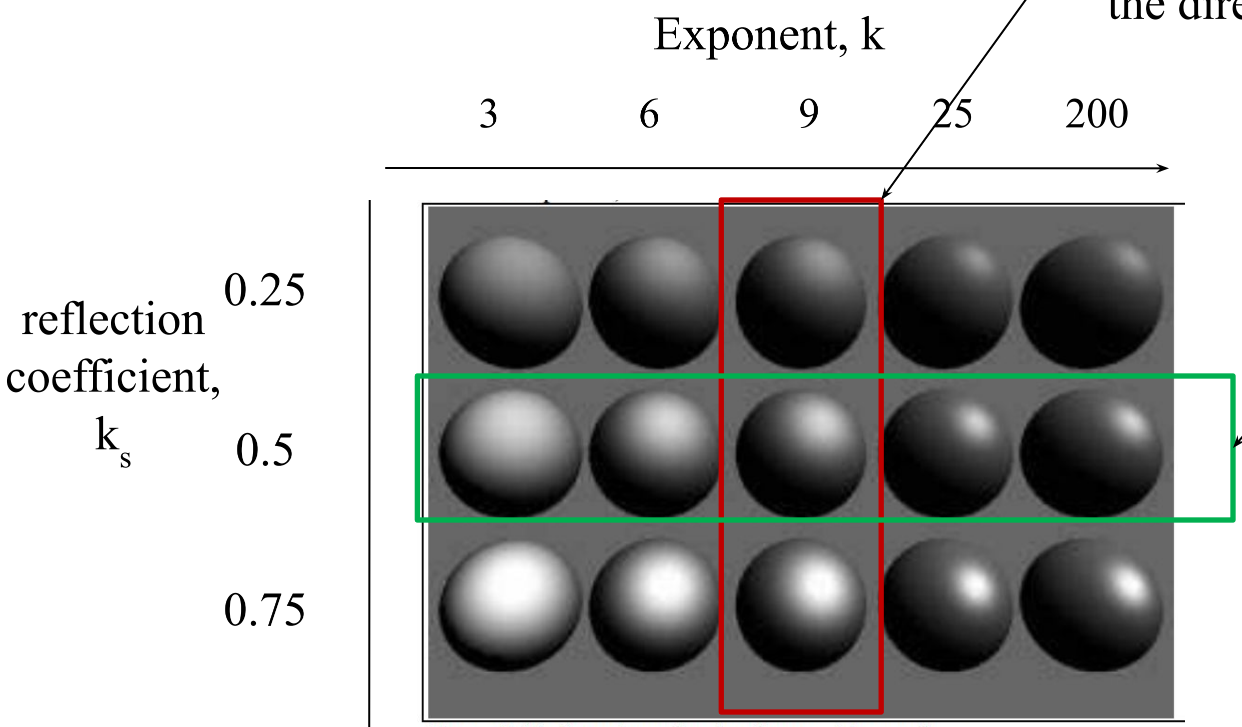
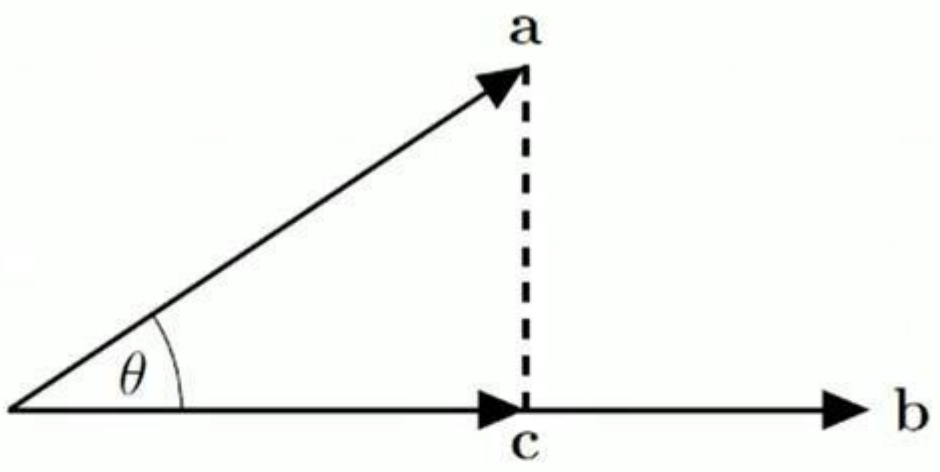


Figure 8.15. Specular reflection from a shiny surface.

More shiny and more highly concentrated along the direction r

More mirror like, more focused reflection with increasing exponent

Vector Projection



- Using the definition of a dot product

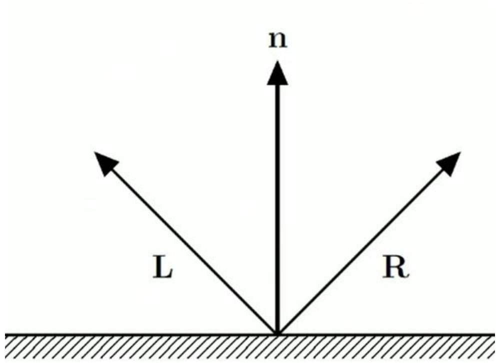
$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}| \cos(\theta) = |\mathbf{a}||\mathbf{b}| \frac{|\mathbf{c}|}{|\mathbf{a}|} = |\mathbf{b}||\mathbf{c}|$$

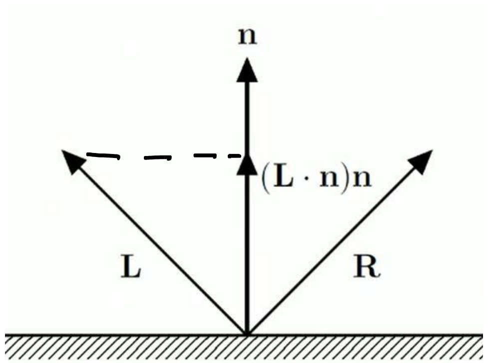
$$\therefore |\mathbf{c}| = \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{b}|}$$

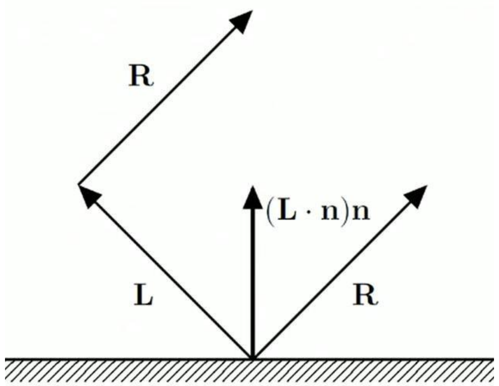
- If $\mathbf{b}$ is a unit vector then $|\mathbf{c}| = \mathbf{a} \cdot \mathbf{b}$ and

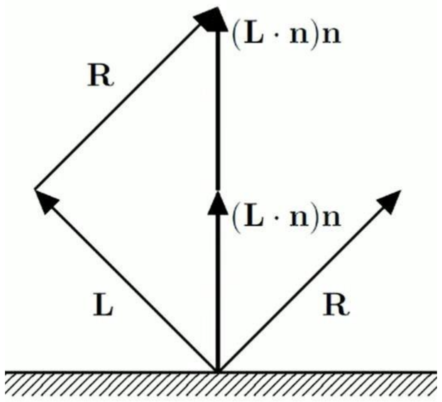
$$\mathbf{c} = (\mathbf{a} \cdot \mathbf{b})\mathbf{b}$$

BRAC UNIVERSITY
Inspiring Excellence





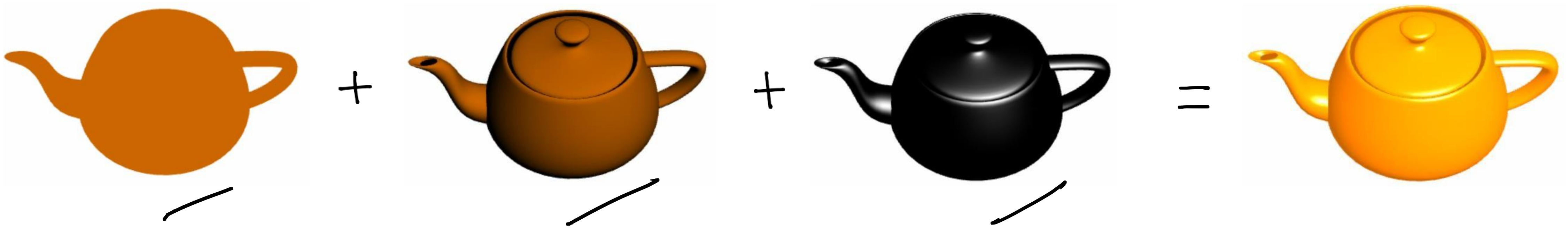




$$\mathbf{L} + \mathbf{R} = 2(\mathbf{L} \cdot \mathbf{n})\mathbf{n}$$

$$\therefore \mathbf{R} = 2(\mathbf{L} \cdot \mathbf{n})\mathbf{n} - \mathbf{L}$$

Phong's Reflection Model



$$I = \underline{I_a K_a} + \underline{I_p K_d \max(L \cdot n, 0)} + \underline{I_p K_s (\max(V \cdot R, 0))^n}$$

Mathematical Calculation of Phong Model

We need to ircorporate the effect light present in environment. Thus total refelcted light also includes ambient component.

Ambient Component = $I_a k_a$

Total reflected light intensity from Q,

I = Ambient Component + Diffuse Component + Specular Component

$= I_a k_a + I_p k_d (\mathbf{L}.\mathbf{N}) + I_p k_s (\mathbf{R}.\mathbf{V})^k$

More specifically,

$$I = I_a k_a + I_p [k_d \max\{(\mathbf{L}.\mathbf{N}),0\} + k_s \max\{(\mathbf{R}.\mathbf{V})^k ,0\}]$$

BRAC UNIVERSITY

Inspiring Excellence

Additional Issues

When there are n light sources in the scene, their effects are cumulative: Intensity at Q,

$$I = I_a k_a + \sum_{(i=1 \text{ to } n)} I_{pi} \{k_d (\mathbf{L}.\mathbf{N}) + k_s (\mathbf{R}.\mathbf{V})^k \}$$

The intesnsity of red, green and blue component of reflected light,

$$I_r = I_a k_{ar} + I_p k_{dr} (\mathbf{L}.\mathbf{N}) + I_p k_s (\mathbf{R}.\mathbf{V})^k$$

$$I_g = I_a k_{ag} + I_p k_{dg} (\mathbf{L}.\mathbf{N}) + I_p k_s (\mathbf{R}.\mathbf{V})^k$$

$$I_b = I_a k_{ab} + I_p k_{db} (\mathbf{L}.\mathbf{N}) + I_p k_s (\mathbf{R}.\mathbf{V})^k$$

k_s : coefficient for specular component which is same as the color of light source, not affected by surface color.

BRAC UNIVERSITY
Inspiring Excellence

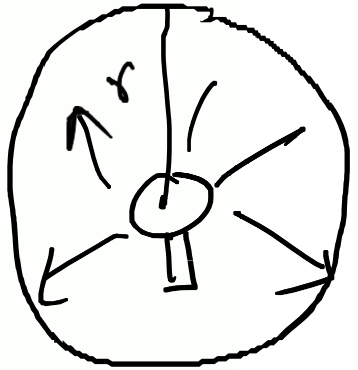
Attenuation

- **Attenuation** is the loss of light energy over space
- In Phong's model attenuation is account for by the variable f_{att} and applied to diffuse and specular components
- Theoretically is should follow the inverse square law, i.e.,

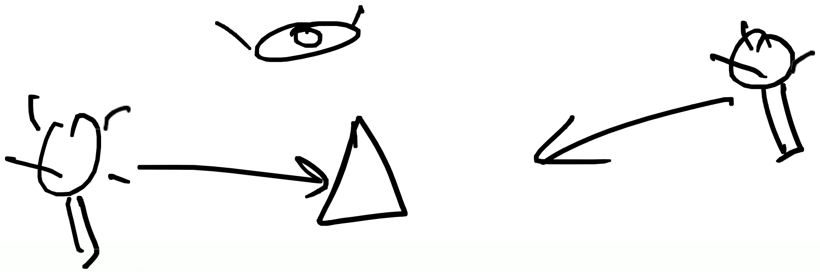
$$f_{att} = \frac{1}{d^2}$$

- In practice this removes too much light, Phong's model uses

$$f_{att} = \max\left[1 - \left(\frac{d}{r}\right)^2, 0\right]$$



where r is the radius of the light source's sphere of influence



- Combining ambient, diffuse, specular and attenuation models results in Phong's model of reflection

$$I = \underline{I_a K_a} + I_p f_{att}\left(K_d \max(L.n, 0) + K_s (\max(V.R, 0))^n\right)$$

- For multiple light sources, the diffuse and specular components are calculated for each light source and added together, i.e.,

$$I = \underline{I_a K_a} + \sum_{i=1}^{m} I_{p_i} f_{att}\left(K_d \max(L_i.n, 0) + K_s (\max(V.R_i, 0))^n\right)$$

Shading

whether an object, or some part of an object is obstructed by another object.

– Z buffer / Depth buffer

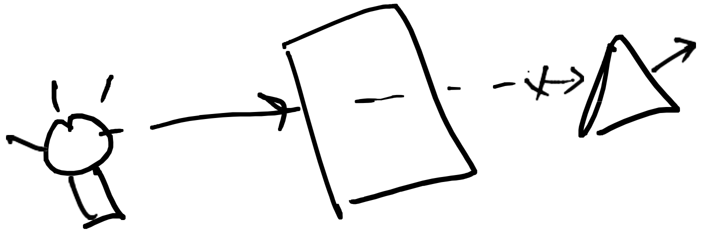


Photo: [Alex Canclini]

Shading

- The process of assigning colors to pixels.

- Shading
  - Flat Shading
  - Smooth Shading
    - Gouraud Shading
    - Phong Shading

Shading Model

- Flat Shading
  - Compute Phong lighting once for entire polygon
- Gouraud Shading
  - Compute Phong lighting at the vertices and interpolate lighting values across polygon
- Phong Shading
  - Interpolate normals across polygon and perform Phong lighting across polygon

Flat Shading

- For each polygon
  - Determines a single intensity value at a chosen point on the polygon
  - Uses that value to shade the entire polygon

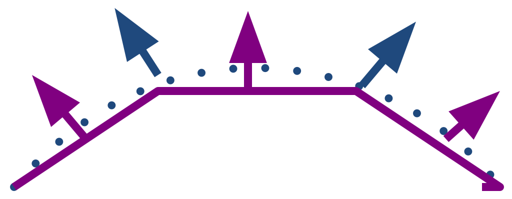
- Assumptions
  - Light source at infinity
  - Viewer at infinity
  - The polygon represents the actual surface being modeled

Problems of Flat Shading

- Specular highlights tends to get lost
- If chosen point on polygon is at location of the light source, then color of the polygon will be significantly distorted.

Smooth Shading

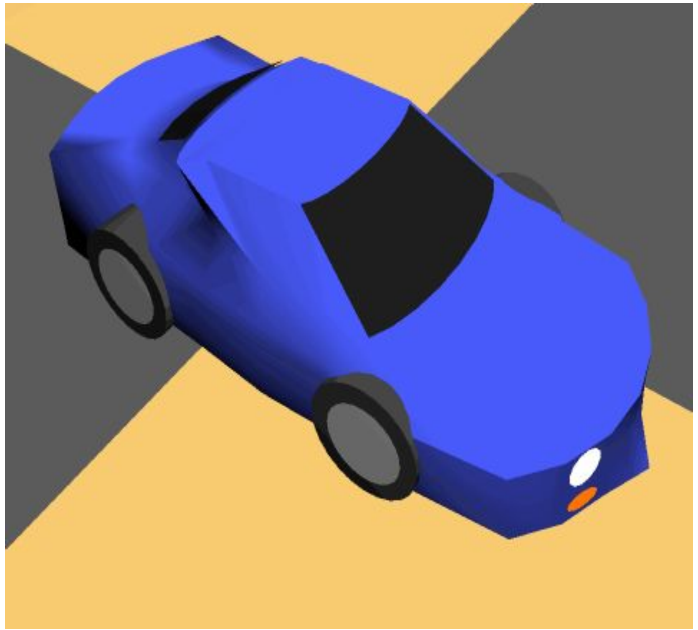
- Introduce vertex normals at each vertex
  - Used only for shading
  - Think of as a better approximation of the real surface that the polygons approximate
  - Finds color value for each point in the polygon individually



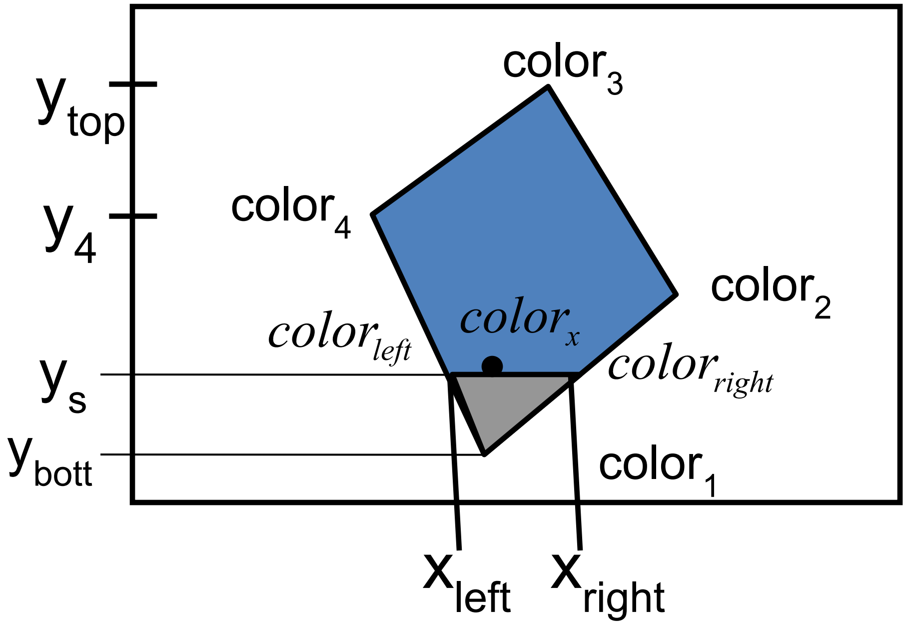
- Two types
  - Gouraud Shading
  - Phong Shading (do not confuse with Phong Lighting Model)

Gouraud Shading

- Most common approach
- Perform Phong lighting at the vertices
- Linearly interpolate the resulting colors over faces
  - Along edges
  - Along scanline



Gouraud Shading

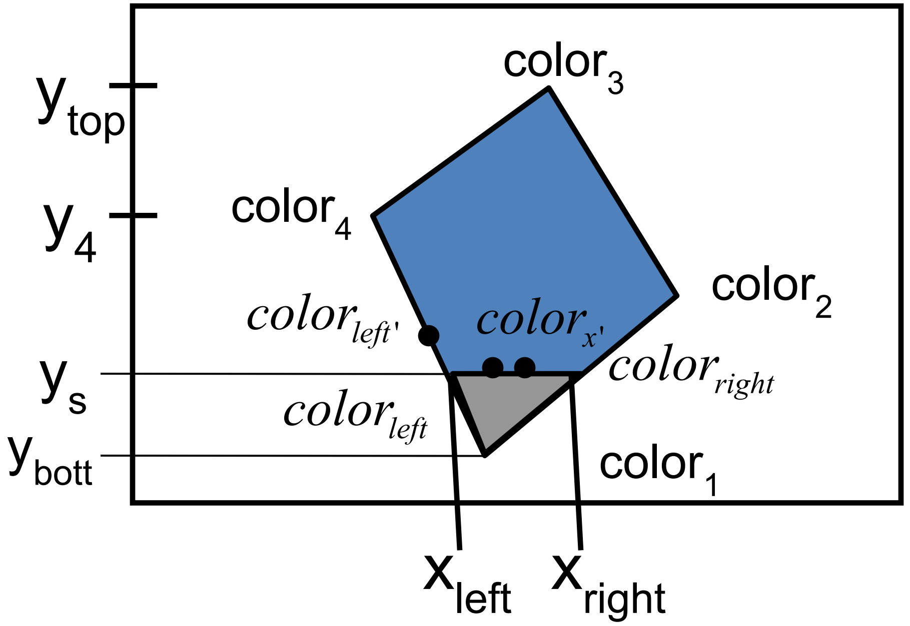


$$color_{left} = color_1 + (color_4 - color_1)\frac{y_s - y_{bott}}{y_4 - y_{bott}}$$

$$color_{right} = color_1 + (color_2 - color_1)\frac{y_s - y_{bott}}{y_2 - y_{bott}}$$

$$color_x = color_{left} + (color_{right} - color_{left})\frac{x - x_{left}}{X_{right} - X_{left}}$$

Gouraud Shading



$$color_x = color_{left} + (color_{right} - color_{left})\frac{x - x_{left}}{x_{left} - x_{right}}$$

$$color_{x'} = color_x + K\Delta x$$

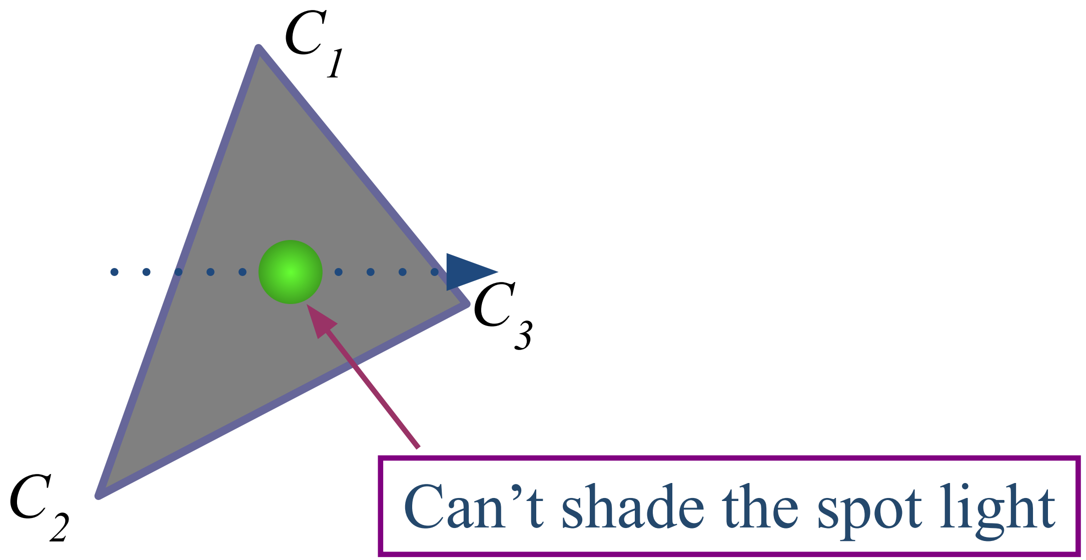
$$color_{left'} = color_{left} + K'\Delta y$$

Calculate the surface normals along the scan line and the edge using incremental approach

BRAC UNIVERSITY
Inspiring Excellence

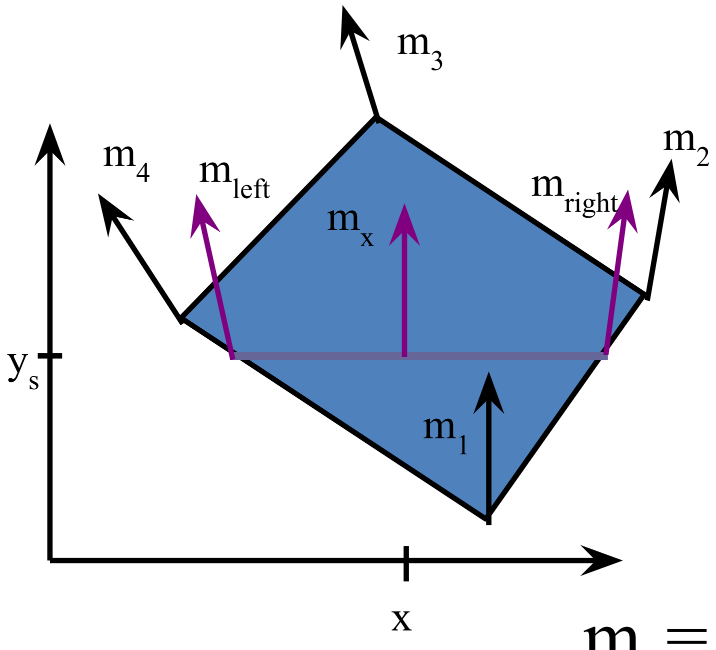
Problem of Gouraud Shading

- Often appears dull
- Lacks accurate specular component



Phong Shading

Interpolate normal vectors of face vertices at each pixel, then perform Phong lighting at each pixel.



$$m_{left} = m_1 + (m_4 - m_1)\frac{y_s - y_1}{Y_4 - y_1}$$

$$m_{right} = m_1 + (m_2 - m_1)\frac{y_s - y_1}{Y_2 - y_1}$$

$$m_x = m_{left} + (m_{right} - m_{left})\frac{x - x_{left}}{x_{right} - x_{left}}$$

Calculate the surface normals along the scan line and the edge using incremental approach

BRAC UNIVERSITY Inspiring Excellence

Phong vs Gouraud Shading

- Phong shading is more smooth
- If a highlight does not fall on a vertex, Gouraud shading may miss it completely, but Phong shading does not.