Illumination

Basic Terms

- Lighting: the process of computing the luminous intensity (i.e., outgoing light) at a particular 3-D point, usually on a surface.
- Shading: the process of assigning colors to pixels.
- Illumination: the transport of energy from light sources to surfaces & points
  - Local illumination
  - Global illumination

Components of Illumination

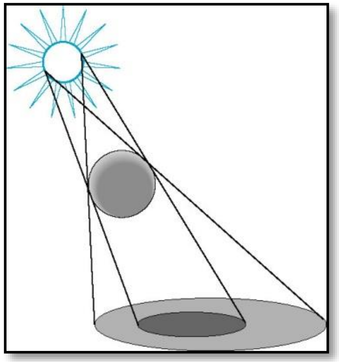
- Two components of illumination: **light sources** and **surface properties**
- Types of Light Sources:
  - Ambient light: no identifiable source or direction
  - Diffuse light - Point: given only by point
  - Diffuse light - Direction: given only by direction
  - Spot light: from source in direction
    - Cut-off angle defines a cone of light
    - Attenuation function (brighter in center)
  - Light source described by a luminance/ intensity ‘I’
    - Each color is described separately
    - I = [I r I g I b] T (I for intensity)

Ambient Light

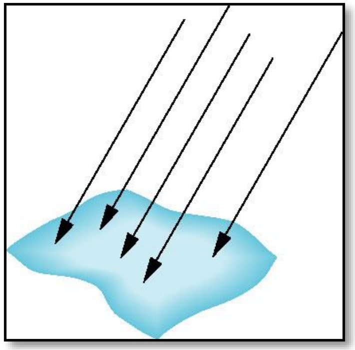
- Global ambient light
  - Independent of light source
  - Lights entire scene
- Local ambient light
  - Contributed by additional light sources
  - Can be different for each light and primary color
- Computationally inexpensive

Diffuse Light

- **Point Source**
  - Given by a point
  - Light emitted equally in all directions
  - Intensity decreases with square of distance
  - Point source $[x\ y\ z\ 1]^T$
  - At point p, intensity received



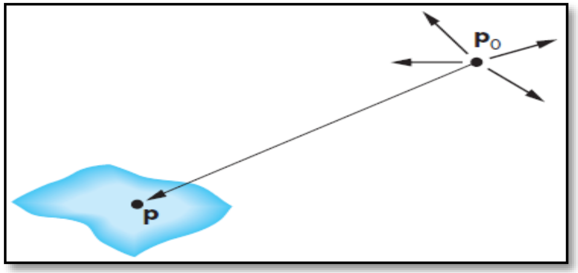
- **Directional Source**
  - Given by a direction
  - Simplifies some calculations
  - Intensity dependents on angle between surface normal and direction of light
  - Distant source $[x\ y\ z\ 0]^T$



Spot Light

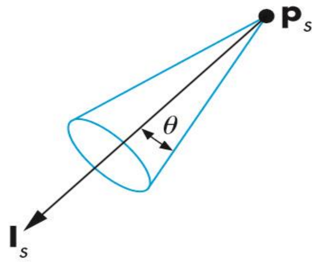
- *Spotlights* are point sources whose intensity falls off directionally.
  - Requires color, point, direction, falloff parameters
- At point p, intensity received

$$\mathrm{i}(\mathbf{p}, \mathbf{p}_0) = \frac{1}{|\mathbf{p} - \mathbf{p}_0|^2} \mathrm{I}(\mathbf{p}_0).$$



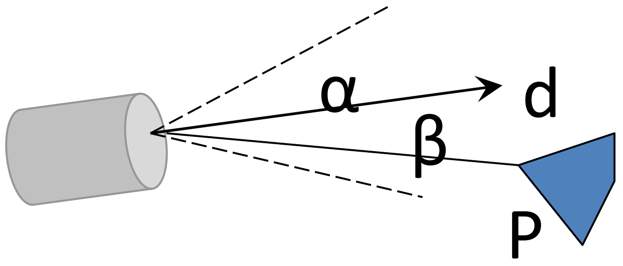
Spot Light

- Source emits light in restricted set of directions,
  - usually direction boundary forms a cone shape.
  - Here θ is Cutoff Cone. No light is seen at points lying outside Cutoff angle.



- Consider, a spot light aimed at direction d.
  - β is the angle between aimed direction d and a line from source to object P
  - Intensity of spot light at P is

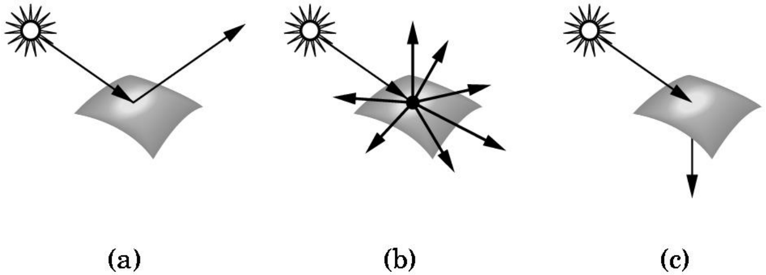
$$\boxed{\text{Intensity at P} = I \cos^{\varepsilon}(\beta)}$$



Types of surface

Interaction betweenlight and materialcan be classified as

- **specular surfaces** – ideal mirror
- **diffuse surfaces** – reflected light is ideally reflected to all directions uniformly
- **translucent surfaces** – allow some lights to penetrate the surface – refraction – glass, watter

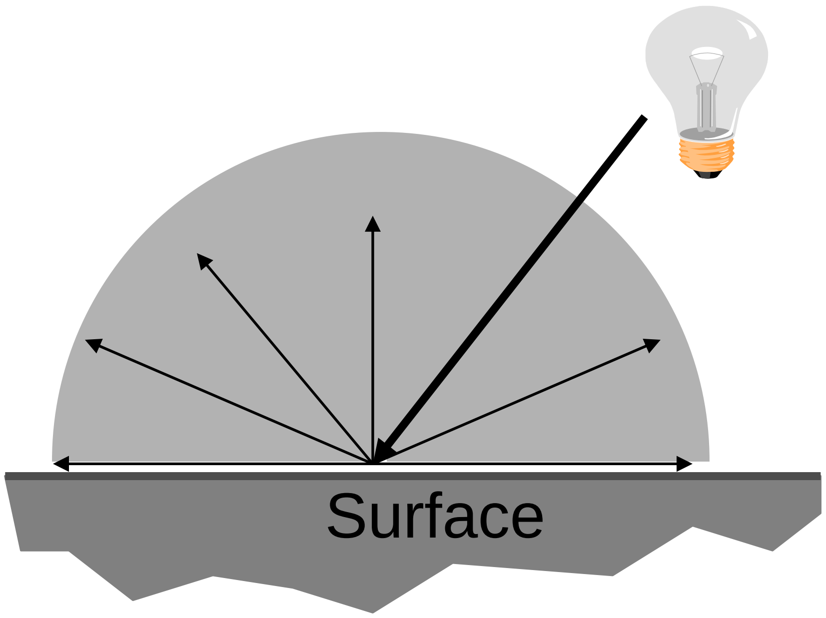


Reflection

Defuse Reflectance, Specular Reflectance

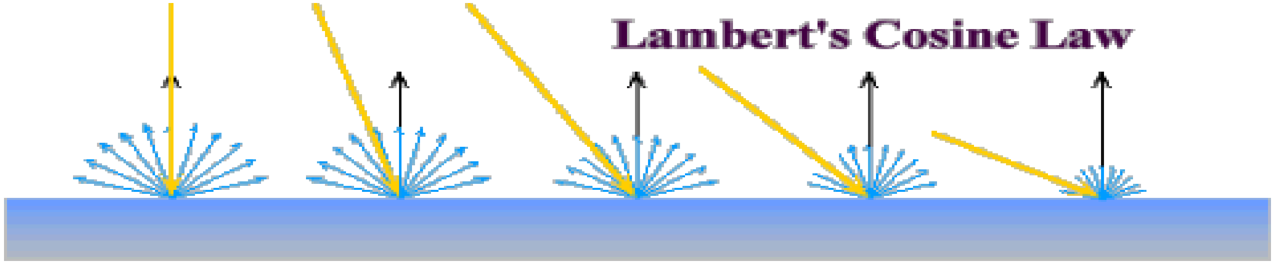
Ideal Diffuse Reflectance

- Assumes surface reflects equally in all directions.
- An ideal diffuse surface is, at the microscopic level, a very rough surface.
  - Example: chalk, clay, some paints



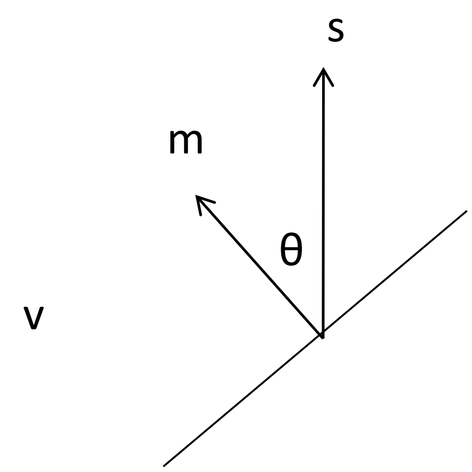
Ideal Diffuse Reflectance

- Ideal diffuse reflectors reflect light according to Lambert's cosine law.
- Lambert's law determines how much of the incoming light energy is reflected. The reflected intensity is independent of the viewing direction.
- But reflected light intensity depends on incident angle of light.



Ideal Diffuse Reflectance

- Suppose light is incident on diffuse surface S.
  - m = direction of Normal of the surface
  - v = viewer's direction
  - s = direction of incident light
  - θ = incident angle
  - I_s = Intensity of Light Source
  - Intensity of reflected light, I_d

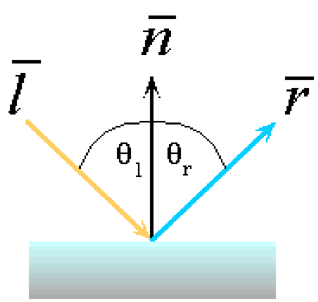


– $I_d = I_s\, Þ_d\, (s.m/|s||m|)$, where $Þ_d$ is *the diffuse reflectance coefficient*.

– If θ is negative, then $I_d = \max(I_s\, Þ_d\, (s.m/|s||m|)\, ,0)$

Ideal Specular Reflection

- Reflection is only at mirror angle. An ideal mirror is a purely specular reflector.
  - View dependent reflection. That is, reflected light's intensity varies with viewr's position.
  - Intensity of refelcted light is strong near mirror angle and strongest at mirror angle.
  - An Ideal specular reflection follows Snell's Law.

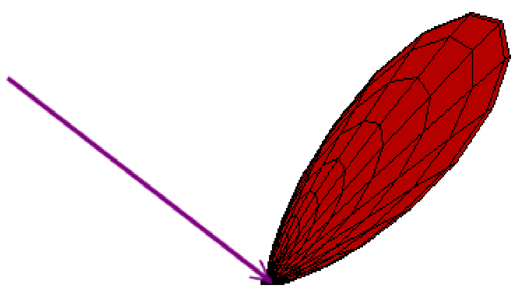


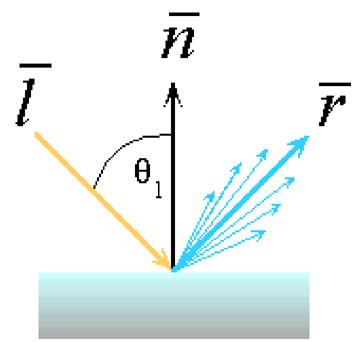
Snell's Laws:

- The incoming ray and reflected ray lie in a plane with the surface normal
- The angle that the reflected ray forms with the surface normal equals the angle formed by the incoming ray and the surface normal.

Non-ideal Reflectors

- Simple Empirical Model:
  - We expect most of the reflected light to travel in the direction of the ideal ray.
  - However, because of microscopic surface variations we might expect some of the light to be reflected just slightly offset from the ideal reflected ray.
  - As we move farther and farther, in the angular sense, from the reflected ray we expect to see less light reflected.



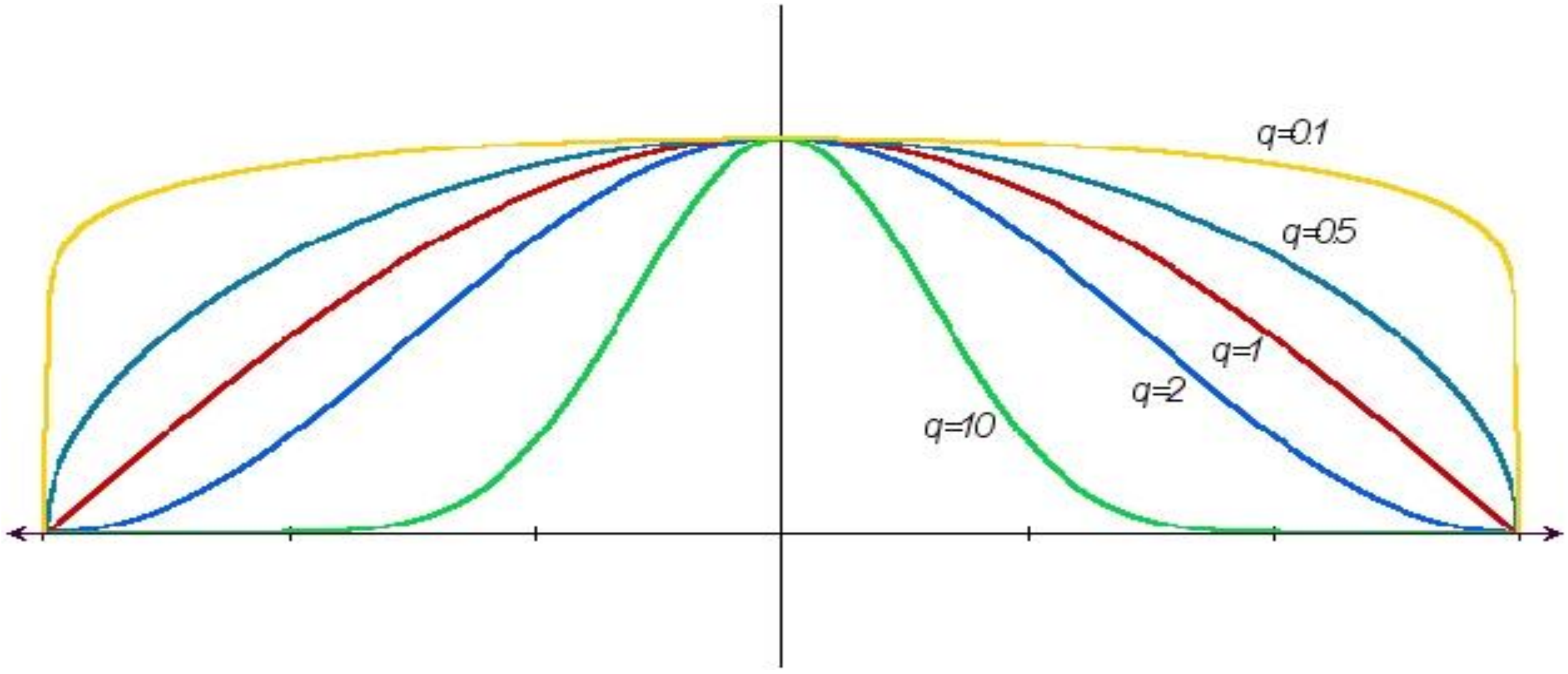


Phong Model

- One function that approximates this fall off is called the ***Phong Illumination*** model.
- This model has no physical basis, yet it is one of the most commonly used illumination models in computer graphics.
- The cosine term is maximum when the surface is viewed from the mirror direction and falls off to 0
- when viewed at 90 degrees away from it. The scalar n_{shiny} controls the rate of this fall off.

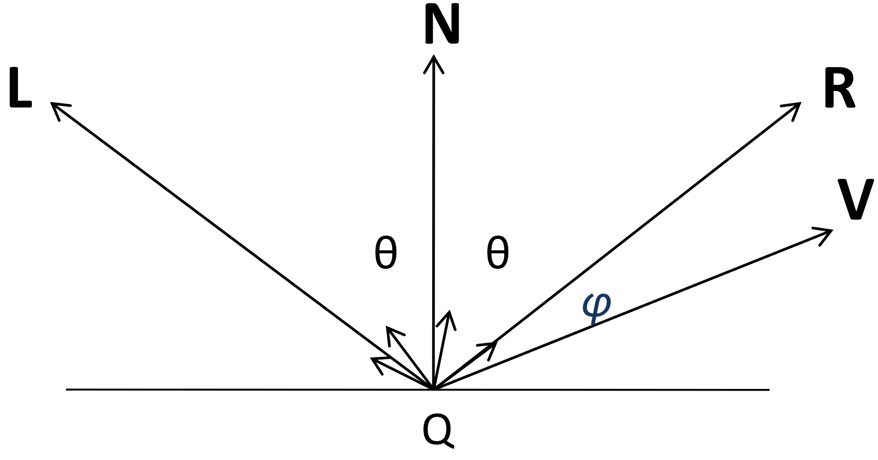
The Phong Model

- Effect of the n_{shiny} coefficient



Phong Model

- Also called Local Illuminatin model as its main focus is on the direct impact of light coming from a source.
- Also models secondary effect of light.
- *Consider a point light source p and viewpoint v. What should be color of light reflected into viewer's eye from Q?*



Mathematical Calculation of Phong Model

- First we consider, 2 extreme cases of light reflection.
  - Diffuse Reflection : $I_d \propto \cos\theta$

    $= I_p \, k_d \cos\theta$

    $= I_p \, k_d \, (\mathbf{L}.\mathbf{N})$

  - Specular Reflection: $I_s \propto (\cos\phi)^k$

    $= I_p \, k_s \, (\cos\phi)^k$

    $= I_p \, k_s \, (\mathbf{R}.\mathbf{V})^k$

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,

$$I = \text{Ambient Component} + \text{Diffuse Component} + \text{Specular Component}$$

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

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

Additional Issues

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

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

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

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

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

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

  – k_s is same as color of specular component is same as color of light source, not affected by surface color.

How to get vector **R** ?

- Find a formula to compute R, the reflction of vector L with respect to normal vector?

R = 2 (**L**.**N**)**N** – **L**

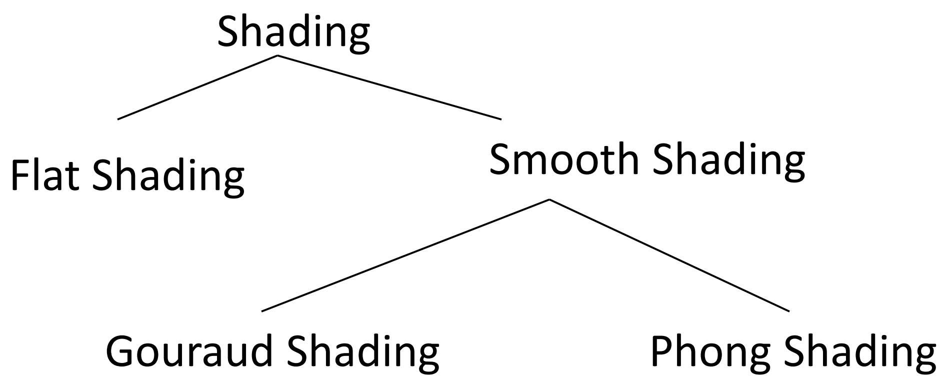
(see Solved Problem 11.10 in Schaum(2nd edition))

Blinn and Torrence Variation

- Calculation of **R** is computaionally expensive. So in phong model the term **R**.**V** is somtimes replaced by **N**. **H** , where **H** is a unit vector that bisect the angle between **L** and **V**.
  - angle between **N** and **H** measures the falloff of intensity.
  - Though calculation of N.H is is computaionally inexpensive relative to R.V, but N.H is not always equal to R.V . In that case calculation of specular component will be approximate. (see Solved Problem 11.11 in Schaum(2nd edition)))

Shading

- The process of assigning colors to pixels.



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

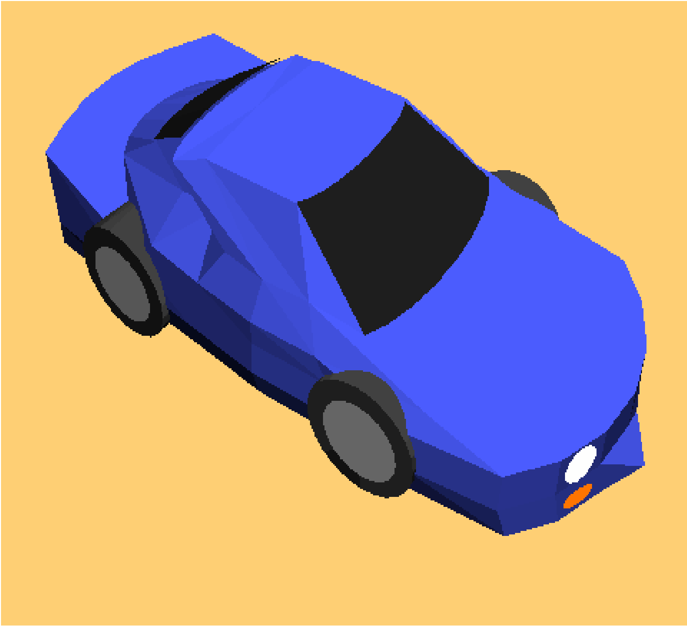
Flat Shading

- 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

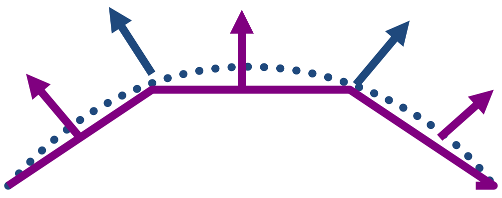
Problems of Flat Shading

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



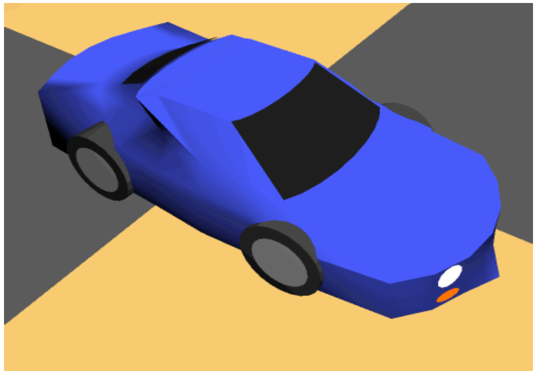
Smooth Shading

- Introduce vertex normals at each vertex
  - Usually different from facet normal
  - 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)

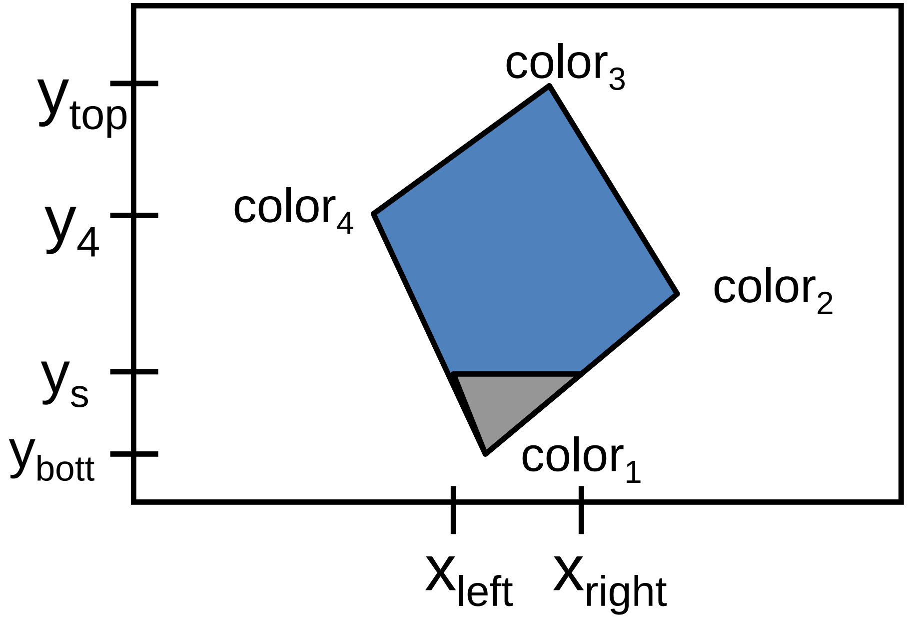


Gouraud Shading

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



Gouraud Shading



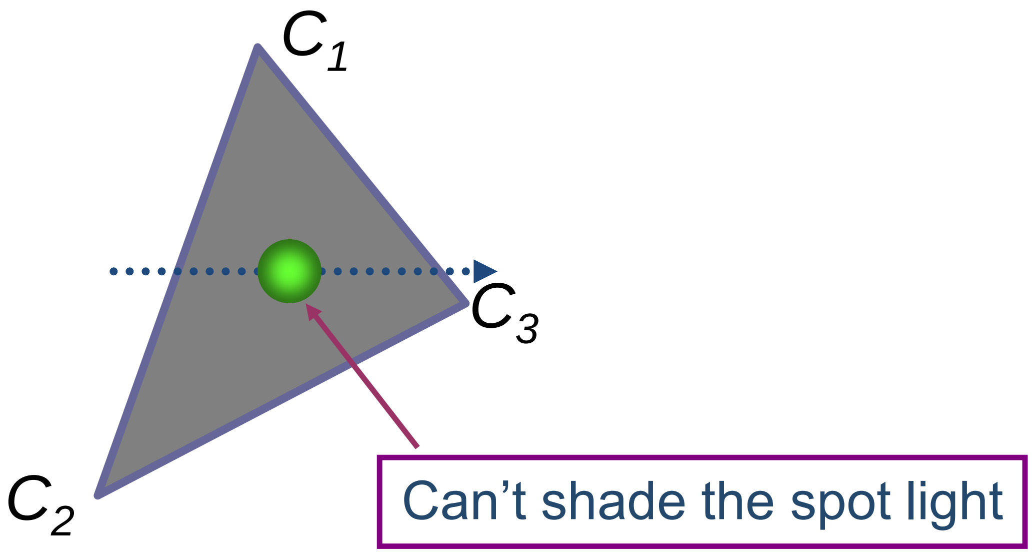
$$color_{left} = color_1 + (color_4 - color_1)\frac{y_s - y_{bott}}{y_4 - y_{bott}}$$

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

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

Problem of Gouraud Shading

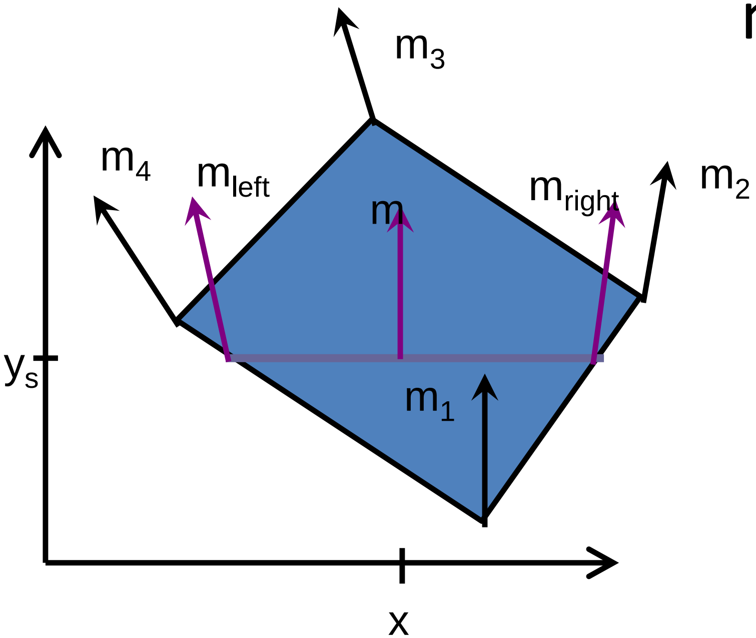
- Often appears dull
- Lacks accurate specular component



Phong Shading

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

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



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

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

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.