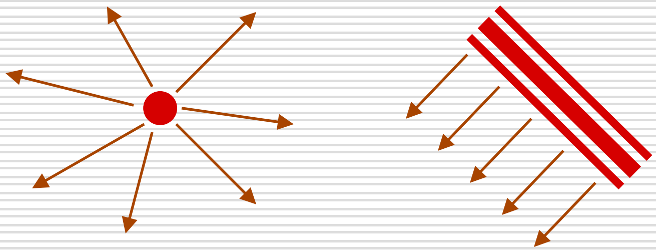
Computer Graphics (L13)
EG678EX

Illumination Models & Surface Rendering Methods

Background

- Perspective Projection + Natural Lighting Effects to Visible Surface → Realistic Effect in Scene
- 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

Light Sources



- Luminous objects
  - Light Emitting Sources
  - Reflecting Sources
- Point Source: Simplest model of light emitter
- Distributed Light Source: modeled as accumulated illumination effects of the points over the surface of the source

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:

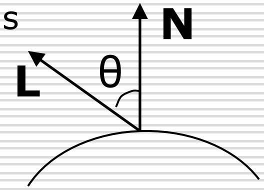
$$I_{ambdiff} = k_d I_a$$

  - 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

$$I_{l,diff} = k_d I_l \cos\theta$$

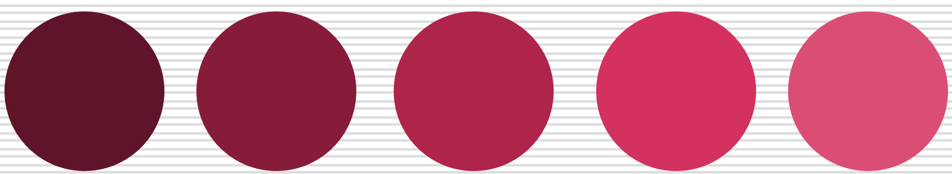
$$I_{l,diff} = k_d I_l (\mathbf{N}.\mathbf{L})$$

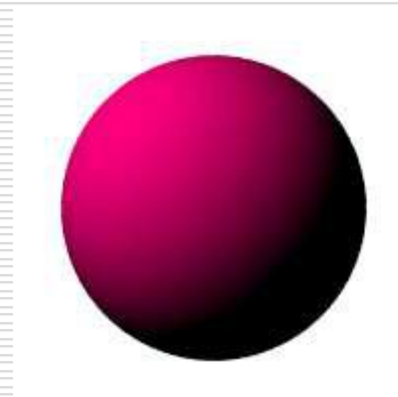
Where **N** and **L** are unit vectors



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

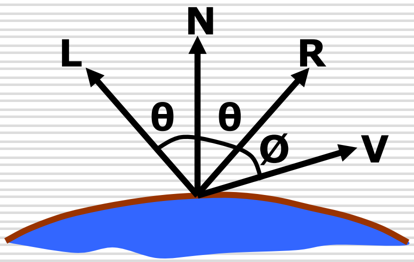




Specular Reflection

- 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

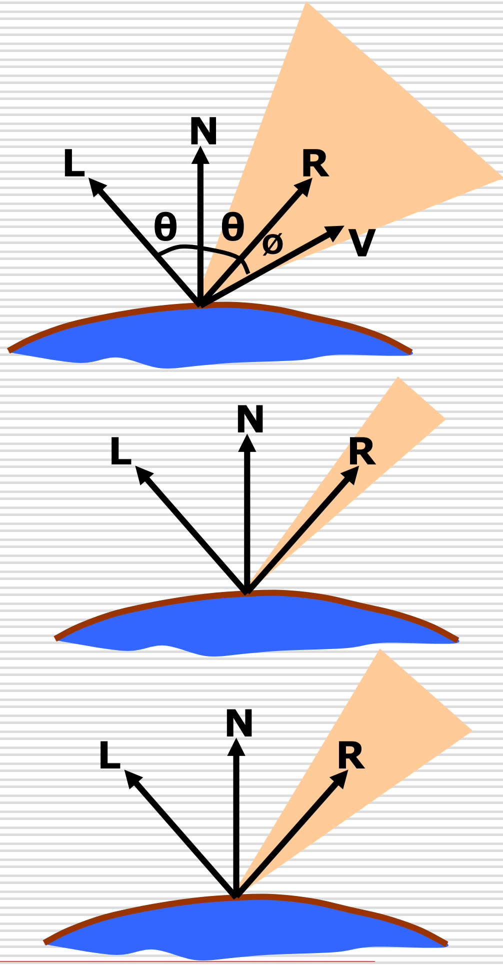
- Intensity of specular reflection: proportional to

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

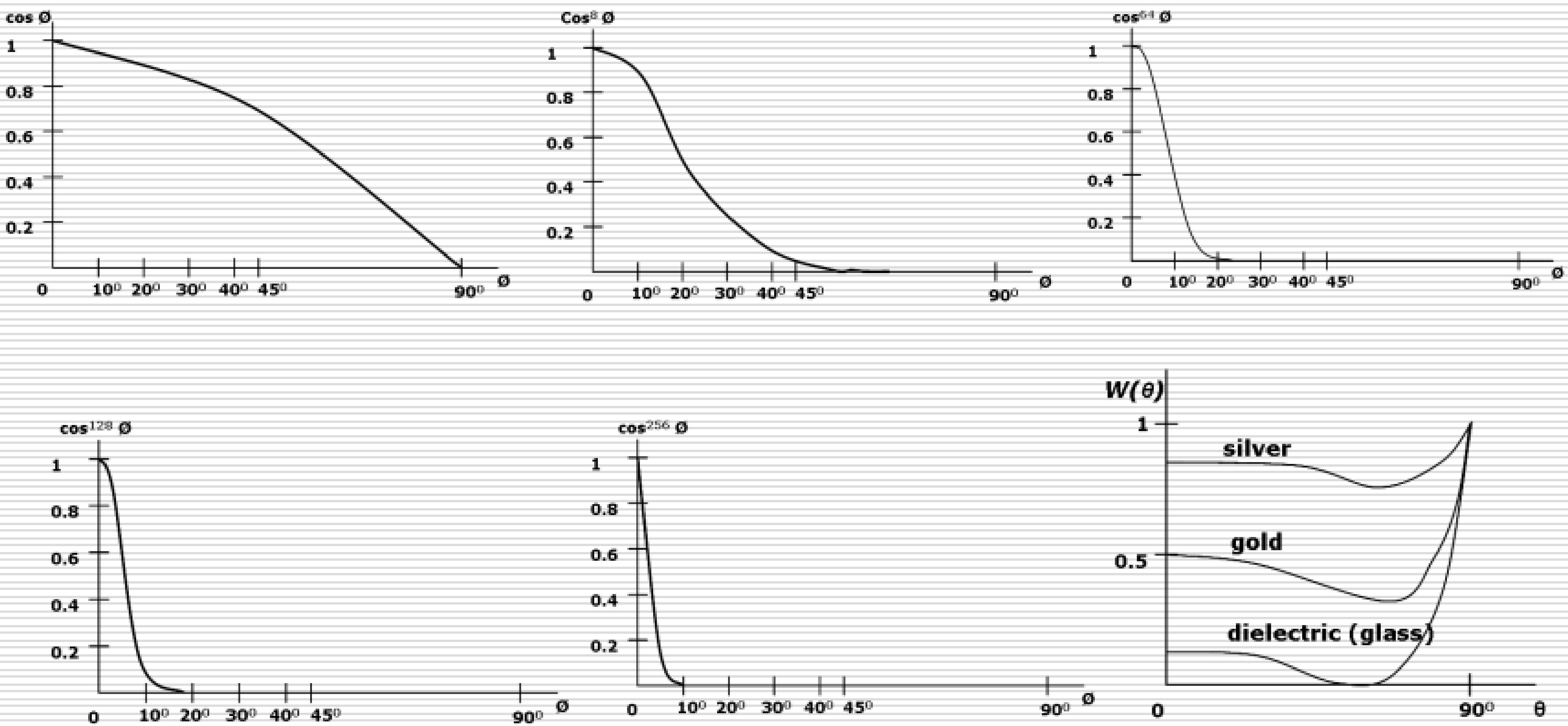
  - n_s → **specular reflection parameter** (depends on surface)
  - Ø ranges from 0 to 90^0 (i.e *cos Ø* varies from 0 to 1)
- Intensity of specular reflection depends on:
  - Material properties of surface
  - Angle of incidence θ
  - Other factors such as polarization and color of the incident light
- 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...)

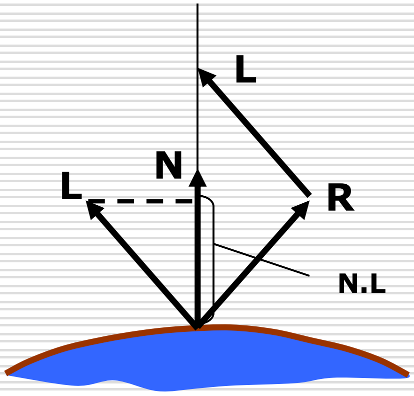
- Simplified form: assume *w(θ) = k_s = constant*

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

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

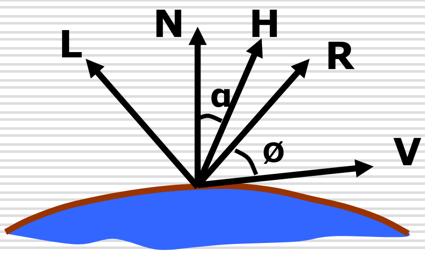
  R + **L** = (2**N**.**L**)**N**

  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:

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

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

- For multiple light sources (n light sources)

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

Warn Model

- Provides method for simulating studio lighting effects
- Light intensities are controlled in different direction
- Light sources are modeled as points on a reflecting surface using the Phong model for the surface points (i.e calculate intensities considering specular and diffuse reflection parameters). Then the intensity in different directions is controlled by selecting values for Phong exponent.
- Light controls used by studio photographers can be simulated

Intensity Attenuation

- The intensity of radiant energy at a point *d* distance far from source is attenuated by $1/d^2$
- Considering intensity attenuation produces realistic lighting effect
  - E.g: if two parallel surfaces with same optical parameters overlap, they would be displayed as one surface
- Using merely $1/d^2$ as attenuation factor for our simple single point light source model, too much intensity variation is produced when *d* is small and a little variation when *d* is large
- Graphical packages have compensated the problem by using inverse linear quadratic function of d for intensity attenuation as:

$$f(d) = \frac{1}{a_0 + a_1 d + a_2 d^2}$$

  - a_0 can be adjusted to prevent f(d) from becoming too large when d is very small
  - Magnitude of attenuation function is limited to 1 as

$$f(d) = \min\left(1, \frac{1}{a_0 + a_1 d + a_2 d^2}\right)$$

- The Phong illumination model considering attenuation is:

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

Color Consideration in Phong Illumination model

- For RGB description, each color in a scene is expressed in terms of R,G and B components
- Various methods:
  - Described by considering the RGB components for e.g. (k_{dR},k_{dG},k_{dB}) of diffuse reflection coefficient vector
    - E.g: For blue light $(k_{dR}=k_{dG}=0)$

$$I_B = k_{aB} I_{aB} + \sum_{i=1}^{n} f(d_i) I_{lBi} \left[k_{dB}(N.L_i) + k_{sB}(N.H_i)^{n_s} \right]$$

  - Described by specifying components of diffuse and specular color vectors for each surface and retaining the reflectivity (k) as a single valued constants

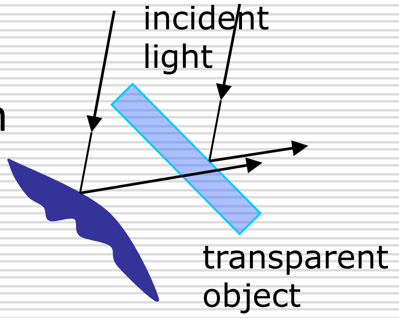
$$I_B = k_a S_{dB} I_{aB} + \sum_{i=1}^{n} f(d_i) I_{lBi} \left[k_d S_{dB}(N.L_i) + k_s S_{sB}(N.H_i)^{n_s} \right]$$

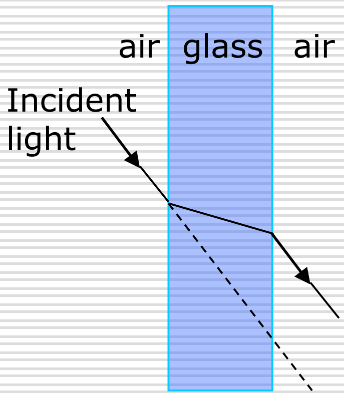
  - Described by specifying wavelength for a color specification. This specification is useful to specify color as more than three components

$$I_B = k_a S_{d\lambda} I_{a\lambda} + \sum_{i=1}^{n} f(d_i) I_{l\lambda i} \left[k_d S_{d\lambda}(N.L_i) + k_s S_{s\lambda}(N.H_i)^{n_s} \right]$$

Transparency

- Transparent surface produces both reflected and transmitted light
- Light intensity depends on relative transparency and position of light source or illuminated object behind or in front of the transparent surface
- To model transparent surface, intensity contribution of light from various sources (illuminated objects) that are transmitted from the surface must be considered in the intensity equation
- Both diffuse and specular reflection take place on transparent surface
- Diffuse effects are important for partially transparent surfaces such as frosted glass
- Diffuse refraction can be generated by decreasing the intensity of the refracted light and spreading intensity contributions at each point on the refracting surface onto a finite area to obtain blurred image of background surface





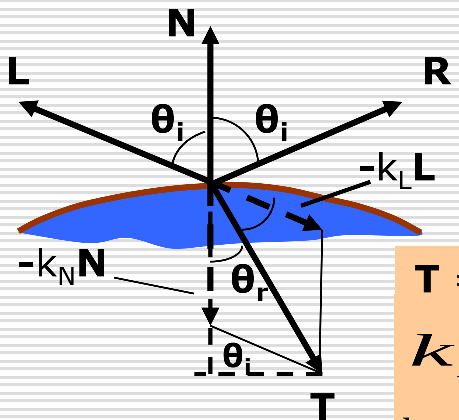
Transparency (contd…)

- The Snell's law is used to calculate the refracted ray direction

$$\sin\theta_r = \frac{\eta_i}{\eta_r}\sin\theta_i$$

- The unit vector along the transmitted light path is given by

$$T = \left(\frac{\eta_i}{\eta_r}\sin\theta_i - \cos\theta_r\right)N - \frac{\eta_i}{\eta_r}L$$



$T = -k_L L - k_N N$

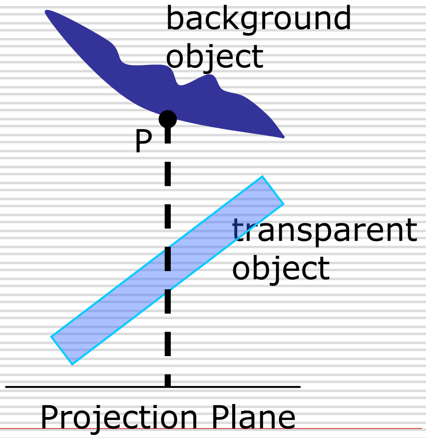
$$k_L = \frac{\eta_i}{\eta_r}$$

$$k_N = \cos\theta_r - k_L\cos\theta_i$$

- Transmitted intensity I_{trans} through a transparent surface from a background object and Reflected intensity I_{refl} from the transparent surface with ***transparency coefficient*** k_t is given by

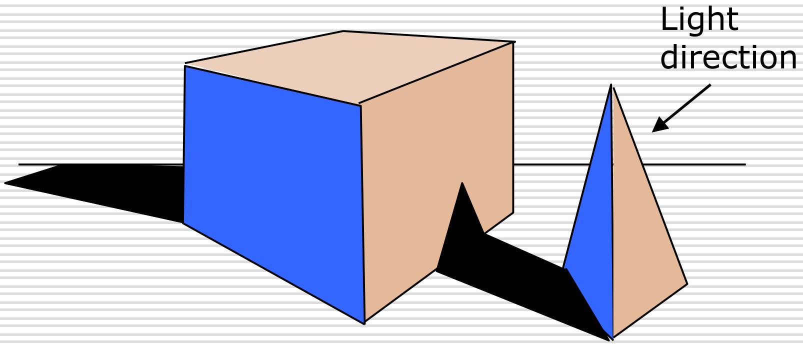
$I = (1-k_t)I_{refl} + k_t I_{trans}$

where $(1-k_t)$ is ***opacity factor***



Shadow

- Hidden surface method with light source at the view position can be used
- The shadow area for all light sources are determined and these shadows could be treated as a surface pattern arrays



Displaying Light Intensities

- The intensity calculated by illumination model must be converted to the allowable intensity of a particular graphics system
- Graphics systems are bilevel (i.e two levels; on and off) and others are capable of displaying several levels
  - For the case of bilevel system, intensities are converted into halftone patterns
- The difference between intensities 0.20 and 0.22 should be perceived same as that of 0.40 and 0.88
- The intensity level in a monitor should be spaced so that the ratio of successive intensities is constant

for n+1 successive intensity levels

$$I_1/I_0=I_2/I_1=\ldots\ldots.I_n/I_{n-1} = r$$

$$I_k = r^k I_0$$

also $I_n = 1 \rightarrow r = (1/I_0)^{1/n}$

so, $I_k = I_0^{(n-k)/n}$

- The lowest intensity value I_0 depends on the characteristics of the monitor (I_0 ranges from 0.005 to around 0.025)
  - the black level displayed on monitor will have some intensity
- The highest intensity value is 1
- For color system (blue color for example)

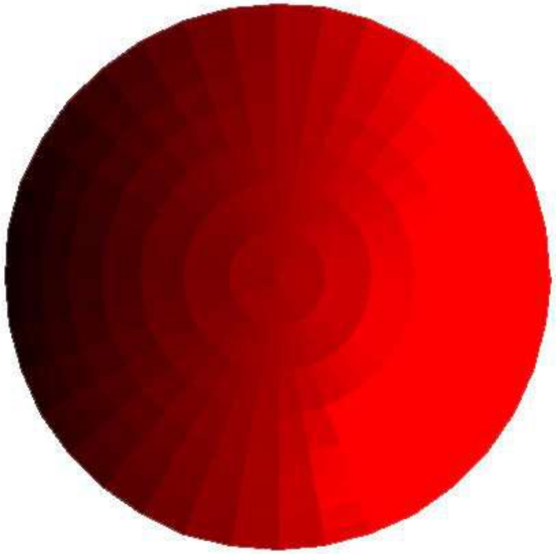
$$I_{Bk} = r_B^k I_{B0}$$

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

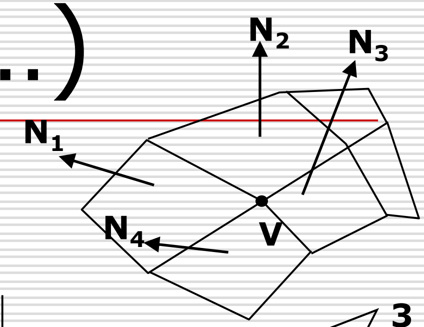


Credit goes to the students

Gouraud Shading

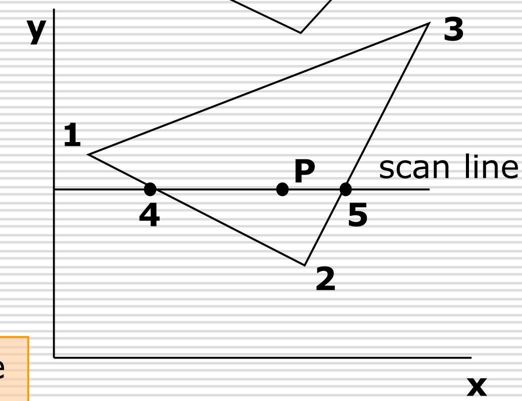
- Calculation Steps:
  - Determine the average unit normal vector at each polygon vertex
  - Calculate each of the vertex intensities by applying an illumination model
  - Linearly interpolate the vertex intensities over the polygon surface
- Intensity discontinuity at the edges of polygons is eliminated
- 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

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

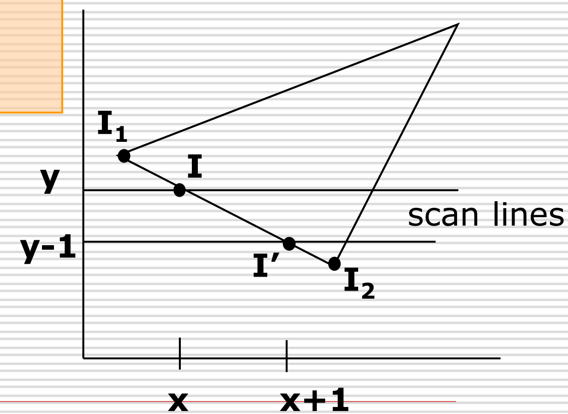


- Intensity interpolation:
  - Along the polygon edges are obtained by interpolating intensities at the edge ends

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

Recursive calculation along the edge

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



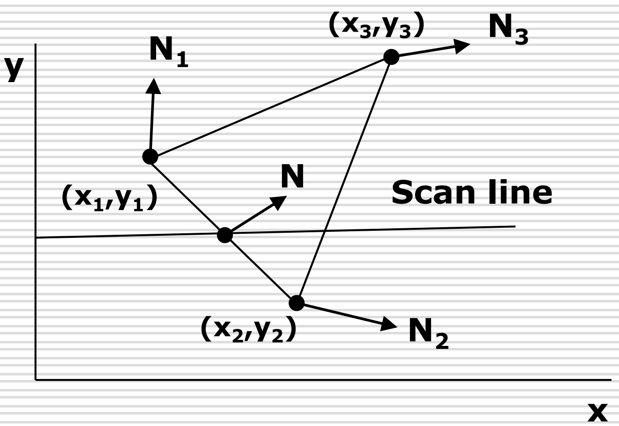
  - Along the scan line between the polygon edges are obtained by interpolating intensities at the intersection of scan line and polygon edges

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

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