Unit 6
Light, Color and Shading

Illumination and Surface Rendering

- Realistic displays of a scene are obtained by perspective projections and applying natural lighting effects to the visible surfaces of object.
- An illumination model is also called lighting model and sometimes called as a shading model which is used to calculate the intensity of light that we should see at a given point on the surface of an object.
- A surface-rendering algorithm uses the intensity calculations from an illumination model.

Light Sources

Sometimes light sources are referred as light emitting object and light reflectors. Generally light source is used to mean an object that is emitting radiant energy e.g. Sun.

Point Source: Point source is the simplest light emitter e.g. light bulb.

Distributed light source: Fluorescent light

Fig: Diverging ray paths from the Point light source

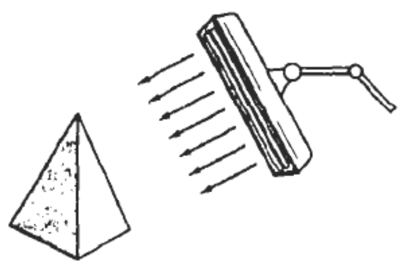
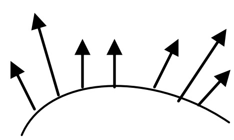
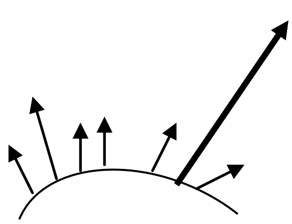


Fig: An object illuminated with a distributed light source

- When light is incident on an opaque surface, part of it is reflected and part of it is absorbed.
- Surface that are rough or grainy, tend to scatter the reflected light in all direction which is called diffuse reflection.



- When light sources create highlights, or bright spots, called specular reflection



Illumination Models

Illumination models are used to calculate light intensities that we should see at a given point on the surface of an object. Lighting calculations are based on the optical properties of surfaces, the background lighting conditions and the light source specifications. All light sources are considered to be point sources, specified with a co-ordinate position and an intensity value (color). Some illumination models are:

1. Ambient Light (Background Light)

- This is a simplest illumination model. We can think of this model, which has no external light source-self-luminous objects. A surface that is not exposed directly to light source still will be visible if nearby objects are illuminated.
- The combinations of light reflections form various surfaces to produce a uniform illumination called ambient light or background light.
- Ambient light has no spatial or directional characteristics and amount on each object is a constant for all surfaces and all directions. In this model, illumination can be expressed by an illumination equation in variables associated with the point on the object being shaded.
- The equation expressing this simple model is

$$I = K_a$$

K_a ranges from 0 to 1.

where I is the resulting intensity and K_a is the object's intrinsic intensity.

- If we assume that ambient light impinges equally on all surfaces from all direction, then

$$I = I_a K_a$$

where I_a is intensity of ambient light. The amount of light reflected from an object's surface is determined by K_a, the ambient-reflection coefficient.

2. Diffuse Reflection

Objects illuminated by ambient light are uniformly illuminated across their surfaces even though light are more or less bright in direct proportion of ambient intensity. Illuminating object by a point light source, whose rays enumerate uniformly in all directions from a single point. The object's brightness varies from one part to another, depending on the direction of and distance to the light source.

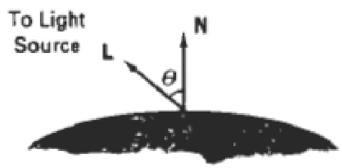
- The fractional amount of the incident light that is diffusely reflected can be set for each surface with parameter K_d, the coefficient of diffuse-reflection.
- Value of K_d is in interval 0 to 1. If surface is highly reflected, K_d is set to near 1. The surface that absorbs almost incident light, K_d is set to nearly 0.
- Diffuse reflection intensity at any point on the surface if exposed only to ambient light is

$$I_{ambdiff} = I_a K_d$$

- Assuming diffuse reflections from the surface are scattered with equal intensity in all directions, independent of the viewing direction (surface called. "Ideal diffuse reflectors") also called Lambertian reflectors and governed by Lambert's cosine law.

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

where I_l is the intensity of the point light source.



It N is unit vector normal to the surface & L is unit vector in the direction to the point slight source, then

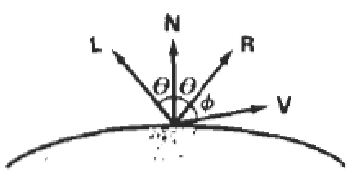
$$I_{l,diff} = K_d I_l (N.L)$$

In addition, many graphics packages introduce an ambient reflection coefficient K_a to modify the ambient-light intensity I_a

$$I_{diff} = K_a I_a + K_d I_l (N.L)$$

3. Specular Reflection and the Phong Model

When we look at an illuminated shiny surface, such as polished metal, a person's forehead, we see a highlight or bright spot, at certain viewing direction. Such phenomenon is called specular reflection. It is the result of total or near total reflection of the incident light in a concentrated region around the specular reflection angle.



N - Unit vector normal to surface at incidence point
R - Unit vector in the direction of ideal specular reflection.
L - Unit vector directed towards point light source.
V - Unit vector pointing to the viewer from surface.
ϕ – Viewing angle relative to the specular reflection direction.

Fig: Specular-Reflection equals the angle of incidence θ

- For ideal reflector (perfect mirror), incident light is reflected only in the specular reflection direction i.e. V and R coincides $(\phi = 0)$.
- Shiny surfaces have a narrow specular-reflection range (narrow ϕ), and dull surfaces have a wider reflection (wider ϕ).
- An empirical model for calculating specular-reflection range developed by Phong Bui Tuong called **Phong-Specular reflection model** (or simply **Phong model**), sets the intensity of specular reflection proportional to $cos^{n_s}\emptyset$ [cos ϕ varies from 0 to 1] where $\mathbf{n_s}$ is a specular reflection parameter.
- Specular reflection parameter n_s is determined by type of surface that we want to display:
  - Very shiny surface: large n_s (say 100 or more) and
  - Dull surface: smaller n_s (down to 1)
  - Perfect reflector: n_s is infinite.

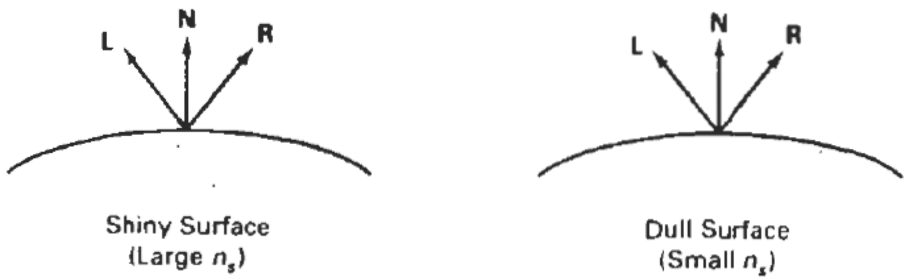


Fig: Modeling specular reflections (shaded area) with parameter $\mathbf{n_s}$

The intensity of specular reflection depends on the material properties of the surface and the angle of incidence (θ), as well as other factors such as the polarization and color of the incident light.

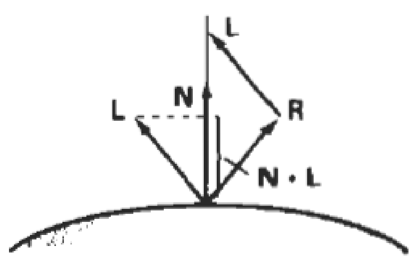
- We can approximately model monochromatic specular intensity variations using a specular-reflection coefficient, W(θ) for each surface over a range θ = *0*° to θ = 90°. In general, W(θ) tends to increase as the angle of incidence increases. At θ = *90*°, W(θ) = 1 and all of the incident light is reflected.
- The variation of specular intensity with angle of incidence is described by ***Fresnel's** laws* of *Reflection*. Using the spectral-reflection function *W(*θ*),* we can write the Phong-Specular-reflection model as:

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

Where I_l is intensity of light source. ϕ is viewing angle relative to SR direction R.

- Transparent materials, such as glass, only exhibit appreciable specular reflections as θ approaches 90°**.** At θ = 0°**,** about 4 percent of the incident light on a glass surface is reflected.
- For many opaque materials, specular reflection is nearly constant for all incidence angles. In this case, we can reasonably model the reflected light effects by replacing W(θ) with a constant specular-reflection coefficient K_s**.**

So, $I_{spec} = K_s I_l cos^{n_s} \emptyset = K_s I_l (V.R)^{n_s}$ Since $\cos\phi$ = V.R



- Vector R in this expression can be calculated in terms of vectors L and N. As seen in Fig. above, the projection of L onto the direction of the normal vector is obtained with the dot product **N.L**. Therefore, from the diagram, we have

R + L = (2N.L)N

And, the specular-reflection vector is obtained as

R = (2N.L)N - L

Polygon-Rendering Methods (Surface-Rendering Methods)

- Application of an illumination model to the rendering of standard graphics objects those formed with polygon surfaces are key technique for polygon rendering algorithm.

- Calculating the surface normal at each visible point and applying the desired illumination model at that point is expensive. We can describe more efficient shading models for surfaces defined by polygons and polygon meshes.
- Scan line algorithms typically apply a lighting model to obtain polygon surface rendering in one of two ways. Each polygon can be rendered with a single intensity, or the intensity can be obtained at each point of the surface using an interpolating scheme.

1. **Constant-Intensity Shading** (Flat-Shading)

The simplest model for shading for a polygon is constant intensity shading also called as Faceted Shading or flat shading. This approach implies an illumination model once to determine a single intensity value that is then used to render an entire polygon. Constant shading is useful for quickly displaying the general appearance of a curved surface.

This approach is valid if several assumptions are true:

a) The light source is sufficiently far so that **N.L** is constant across the polygon face.
b) The viewing position is sufficiently far from the surface so that **V.R** is constant over the surface
c) The object is a polyhedron and is not an approximation of an object with a curved surface.

Even if all of these conditions are not true, we can still reasonably approximate surface-lighting effects using small polygon facets with flat shading and calculate the intensity for each facet, say, at the center of the polygon.

2. **Interpolated Shading**

An alternative to evaluating the illumination equation at each point on the polygon, we can use the interpolated shading, in which shading information is linearly interpolated across a triangle from the values determined for its vertices. Gouraud generalized this technique for arbitrary polygons. This is particularly easy for a scan line algorithm that already interpolates the z-value across a span from interpolated z-values computed for the span's endpoints.

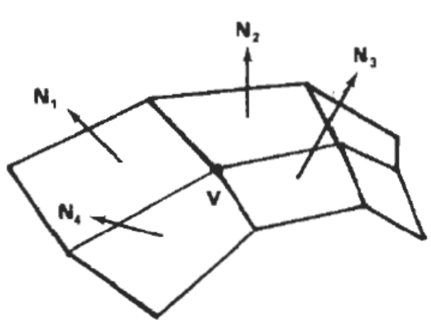
Gouraud Shading

Gouraud shading, also called intensity interpolating shading or color interpolating shading eliminates intensity discontinuities that occur in flat shading. Each polygon surface is rendered with Gouraud shading by performing following calculations.

1. Determine the average unit normal vector at each polygon vertex.
2. Apply an illumination model to each vertex to calculate the vertex intensity.
3. Linearly interpolate the vertex intensities over the surface of the polygon

Step 1: At each polygon vertex, we obtain a normal vertex by averaging the surface normals of all polygons sharing the vertex as:

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



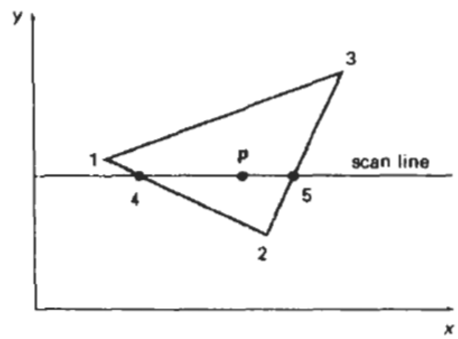
Here in example: $N_v = \dfrac{N_1 + N_2 + N_3 + N_4}{|N_1 + N_2 + N_3 + N_4|}$

Where N_v is normal vector at a vertex sharing 4 surfaces as in figure.

Step 2: Once we have the vertex normals (N_v), we can determine the intensity at the vertices from a lighting model.

Step 3: Now to interpolate intensities along the polygon edges, we consider following figure.

In figure, the intensity of vertices 1, 2, 3 are I_1, I_2, I_3, which are obtained by averaging normals of each surface sharing the vertices and applying a illumination model. For each scan line, intensity at intersection of line with Polygon edge is linearly interpolated from the intensities at the edge end point.



So intensity at point 4 is to interpolate between intensities I_1 and I_2 using only the vertical displacement of the scan line:

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

Similarly, the intensity at point 5 is obtained by linearly interpolating intensities at I_2 and I_3 as

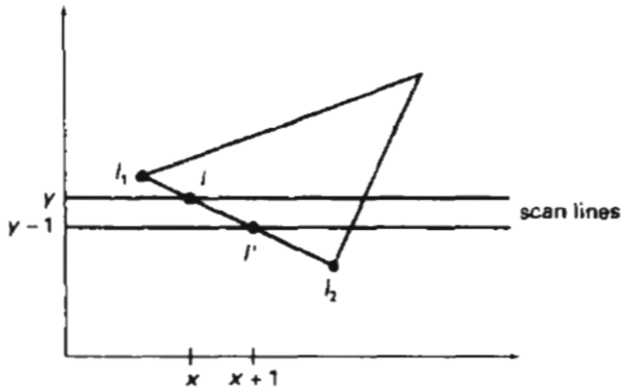
$$I_5 = \frac{y_5 - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_5}{y_3 - y_2} I_2$$

The intensity of a point P in the polygon surface along scan-line is obtained by linearly interpolating intensities at I_4 and I_5 as,

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

Then incremental calculations are used to obtain Successive edge intensity values between scan-lines as and to obtain successive intensities along a scan line. As shown in Fig. below, if the intensity at edge position (x, y) is interpolated as:

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



Then, we can obtain the intensity along this edge for next scan line at y-1 position as

$$I' = \frac{y-1-y_2}{y_1-y_2} I_1 + \frac{y_1-(y-1)}{y_1-y_2} I_2 \qquad = I + \frac{I_2 - I_1}{y_1 - y_2}$$

Similar calculations are made to obtain intensity successive horizontal pixel.

<u>Advantages</u>: Removes intensity discontinuities at the edge as compared to constant shading.
<u>Disadvantages</u>: Highlights on the surface are sometimes displayed with anomalous shape and linear intensity interpolation can cause bright or dark intensity streak called mach-bands.

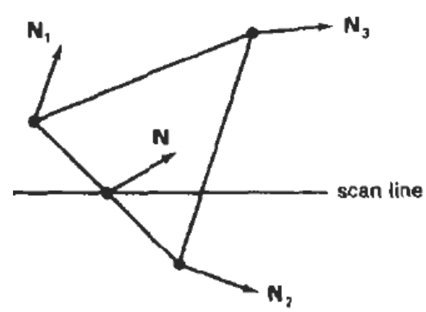
Phong Shading
A more accurate method for rendering a polygon surface is to interpolate normal vector and then apply illumination model to each surface point. This method is called **Phong shading** or **normal vector interpolation shading**. It displays more realistic highlights and greatly reduces the mach-band effect.

A polygon surface is rendered with Phong shading by carrying out following calculations.

- Determine the average normal unit vectors at each polygon vertex.
- Linearly interpolate vertex normals over the surface of polygon.
- Apply illumination model along each scan line to calculate projected pixel intensities for the surface points.

In figure, N_1, N_2, N_3 are the normal unit vectors at each vertex of polygon surface. For scan-line that intersect an edge, the normal vector N can be obtained by vertically interpolating normal vectors of the vertex on that edge as.

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



Incremental calculations are used to evaluate normals between scan lines and along each individual scan line as in Gouraud shading. Phong shading produces accurate results than the direct interpolation but it requires considerably more calculations.

Fast Phong Shading

Surface rendering with Phong shading can be speeded up by using approximations in the illumination-model calculations of normal vectors. Fast Phong shading approximates the intensity calculations using a Taylor series expansion and Triangular surface patches. Since Phong shading interpolates normal vectors from vertex normals, we can express the surface normal N at any point (x, y) over a triangle as:

$$N = Ax + By + C$$

Where A, B, C are determined from the three vertex equations.

$$N_k = Ax_k + By_k + C, \quad k = 1, 2, 3 \text{ for } (x_k, y_k) \text{ vertex.}$$

Omitting the reflectivity and attenuation parameters, we can write the calculation for light-source diffuse reflection from a surface point (x, y) as

$$I_{diff}(x, y) = \frac{L.N}{|L|.|N|} = \frac{L.(Ax + By + C)}{|L|.|Ax + By + C|} = \frac{(L.A)x + (L.B)y + (L.C)}{|L|.|Ax + By + C|}$$

Re writing this,

$$I_{diff}(x, y) = \frac{ax + by + c}{(dx^2 + exy + fy^2 + gx + hy + i)^{\frac{1}{2}}} \quad \text{-------------------- (1)}$$

Where parameters a, b, c, d… are used to represent the various dot products as $a = \frac{L.N}{|L|}$... and so on

Finally, denominator of equation (1) can be expressed as Taylor series expansions and retains terms up to second degree in x and y. This yields

$$I_{diff}(x, y) = T_5x^2 + T_4xy + T_3y^2 + T_2x + T_1y + T_0$$

Where each T_k is a function of parameters a, b, c, d, and so forth.

This method still takes twice as long as in Gouraud shading. Normal Phong shading takes six to seven times that of Gouraud shading.