Unit 8

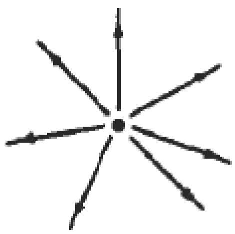
Illumination Models & Surface Rendering Techniques

- Illumination model is used to calculate the intensity of light reflected from a point on a surface.
- Surface rendering uses the intensity calculations from the illumination model to determine the light intensity at all pixels in the image, by possibly, considering light propagation between surfaces in the scene.

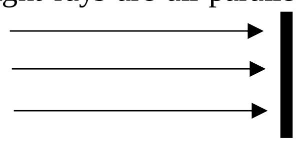
Light sources

Light sources are referred as light emitting object and light reflectors.

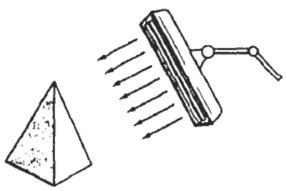
i. ***Point source:*** A point light source emits light equally in all directions from a single point.



ii. ***Parallel source:*** Light rays are all parallel. May be modeled as a point source at infinity (the sun).

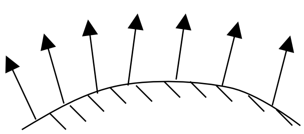


iii. ***Distributed source:*** All light rays orginate at a finite area in space. A nearby sources such as fluorescent light.



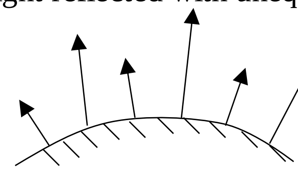
Types of Reflection

i. Diffuse Reflections: Surfaces appear equally bright from all viewing angles since they reflect light with equal intensity in all directions. E.g. rough surfaces



- Position of viewer is not important.

ii. ***Specular reflection:*** Light reflected with unequal intensity. E.g. shiny surface.



- Position of viewer is important.

Illumination model / Lighting model / Shading model

An illumination model is a formula in variables associated to the surface properties and light conditions to calculate the intensity of light reflected from a point on a surface.

Based on standard lighting conditions in a scene, some illumination models are:

i. ***Ambient light:***
Surface getting light from various reflected source i.e. light not coming directly from a light source but coming after getting reflected from other surface.

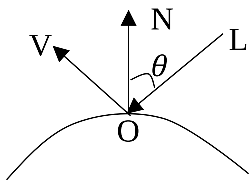
- Ambient light has no spatial or directional characteristics and amount on each object is a constant for all surfaces and all directions.

The reflected intensity I due to ambient light of any point on the surface is:

$$I = I_a K_a$$

Where, I_a= intensity of ambient light.
K_a= ambient reflection coefficient, $0 \leq K_a \leq 1$.

ii. ***Diffuse reflection:***
Light reflected with equal intensity in all direction. Amount of light seen by viewer is independent of viewer direction. Brightness depends only on the angle θ between light direction and the surface normal.



L=light source

N=normal to surface

V=viewer direction

θ=angle bet^n S & N

The reflected intensity I of any point on surface is

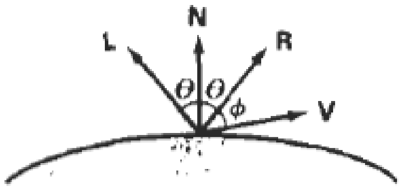
$$I = I_l K_d \vec{N} . \vec{L} = I_l K_d cos\theta$$

Where, K_d= coefficient of diffuse reflection

Net diffuse reflection= Diffuse reflection due to ambient light +Diffuse reflection due to light source = $I_{amb,\ diff} + I_{l,\ diff} = I_a K_a + I_l K_d cos\theta$

iii. ***Specular reflection and Phong model***

In shiny surface we see highlight or bright spot from certain viewing directions called specular reflection. (Light reflected with unequal intensity).



L=incident light

N=normal to surface

R=direction of ideal specular reflection

V=viewer

ϕ=angle bet^n R & V

$$I_{l,spec} = K_s I_l(\vec{V}.\vec{R})^{n_s}$$

Where, K_s= specular reflection coefficient
n_s= specular reflection parameter

Phong model:

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

Combined diffuse and specular reflection

$$I = I_{diff} + I_{spec}$$

$$I = I_a K_a + I_l K_d(\vec{N}.\vec{L}) + K_s I_l(\vec{V}.\vec{R})^{n_s}$$

Intensity Attenuation

Intensity is attenuated by the factor $1/d^2$**,** where d is the distance that the light has traveled. This means that a surface close to the light source (small ***d)*** receives a higher incident intensity from the source than a distant surface (large ***d).***
Graphics package have a general inverse quadratic attenuation function

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

User can fiddle with the coefficients a_0, a_1, a_2 to obtain a variety of lighting effects for a scene. The a_0 can be adjusted to prevent f(d) from becoming too large when d is very small. This is an effective method for limiting intensity values when a single light source is used to illuminate a scene.

With a given set of attenuation coefficients, we can limit the magnitude of the attenuation function to 1 with the calculation

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

Using this function, we can then write our basic illumination model as

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

Where d_i is the distance light has traveled from light source *i*.

Color Considerations

To incorporate color, we need to write the intensity equation as a function of the color properties of the light sources and object surfaces. Diffuse reflection coefficient vector for RGB component (k_{dR}, k_{dG}, k_{dB}).

For blue reflectivity component $k_{dR} = k_{dG} = 0$.

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

Surfaces are typically illuminated with white light sources, and in general we can set surface color so that the reflected light has nonzero values for all three RGB components. Calculated intensity levels for each color component can be used to adjust the corresponding electron gun in an RGB monitor.

In his original specular-reflection model, Phong set parameter Ks to a constant value independent of surface color. This produces specular reflections that are same color as the incident light (usually white), which gives the surface plastic appearance. For a non-plastic material, the color of the specular reflection is a function of surface properties and may be different from the color of the incident light and the color of the diffuse reflections.

Another method diffuse and specular color vector

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

Color specification with its spectral wavelength λ.

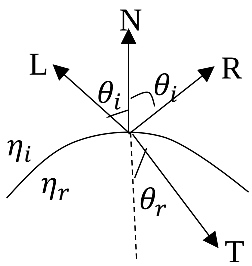
$$I_\lambda = K_a S_{d\lambda} I_{a\lambda} + \sum_{i=1}^{n} f(d_i) I_{L\lambda i} [k_d S_{d\lambda}(N.L_i) + k_s S_{s\lambda}(N.H_i)^{n_s}]$$

Transparency

A transparent surface in general produces both reflected and transmitted light. The relative contribution of the transmitted light depends on the degree of transparency of the surface and whether any light sources are behind the transparent surface. Both diffuse and specular transmission can take place at the surfaces of a transparent object.

According to Snell's Law

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



θ_i=angle of incidence

η_i=index of incidence

θ_r=angle of refraction

η_r=index of refraction

Fig: Reflection direction R and refraction direction T for a ray of light incident **upon** a surface with index of refraction η_r.

Total surface intensity is calculated as

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

Where,

$I_{reflect}$= reflected intensity

I_{trans}= transmitted intensity

k_t= transparency coefficient

$(1 - k_t)$= opacity factor

For highly transparent objects k_t is near 1.

For opaque objects k_t is near 0.

Shadows

Hidden surface method can be used to locate area where light sources produce shadows. By applying a hidden surface method with a light source at the view position, we can determine which surface sections cannot be seen from the light source.

Once we have determined the shadow area for all light sources, the shadows could be treated as surface patterns and store in pattern arrays. Shadow patterns generated by a hidden surface method are valid for any selected viewing position, as long as the light source positions are not changed.

Polygon Rendering Method

a) Constant intensity shading method
b) Gouraud shading method (Intensity Interpolation)
c) Phong shading mehod (Normal vector interpolation)

a) *Constant Intensity Shading (Flat Shading)Method*

- A fast and simple method for rendering an object with polygon surfaces is constant intensity shading.
- In this method, a single intensity is calculated for each polygon and the intensity is applied to all the points of surface of polygon. Hence, all the points over the surface of the polygon are displayed with same intensity value.
- Constant shading is useful for quickly displaying the general appearance of a curved surfaces. This approach is valid if:
    a. Light source is at infinity ($\vec{N} \cdot \vec{L}$ is constant on polygon).
    b. Viewer is at infinity ($\vec{V} \cdot \vec{R}$ is constant on polygon).
    c. Object is polyhedron and is not an approximation of an object with a curved surface.

b) Gouraud shading method

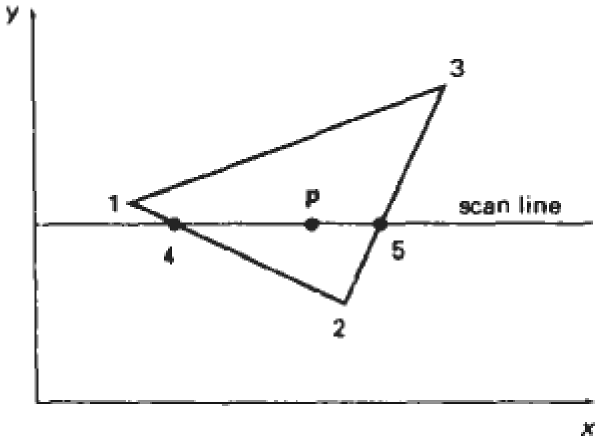
- Developed by Henri Gouraud.
- It renders the polygon surface by linearly interpolating intensity values across the surface.
- Intensity values for each polygon are matched with the values of adjacent polygon along the common edge, thus eliminating the intensity discontinuities that occur in flat shading.

Steps:

1. Determine the average unit normal vector at each vertex of polygon.

$$\vec{N}_{avg} = \frac{\sum_{i=1}^{n} \vec{N}_i}{|\sum_{i=1}^{n} \vec{N}_i|}$$

2. Apply illumination model at each vertex to calculate the vertex intensity.
3. Linearly interpolate the vertex intensities over the projected area of polygon.



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

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

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

Advantages:

- It removes the intensity discontinuity which exists in constant shading model.
- It can be combined with hidden surface algorithm to fill in the visible polygon along each scan line.

Disadvantages:

- Causes bright or dark intensity streak called mach bands.

c) Phong shading method

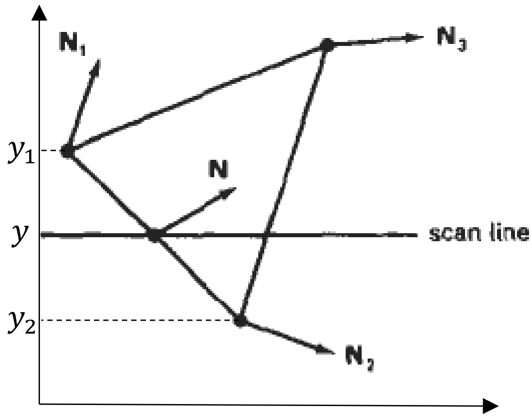
- Developed by Phong Bui Tuong.
- It renders the polygon surface by linearly interpolating normal vector across the surface.

Steps:

1. Determine the average unit normal vector at each vertex of polygon.

$$\vec{N}_{avg} = \frac{\sum_{i=1}^{n} \vec{N}_i}{|\sum_{i=1}^{n} \vec{N}_i|}$$

2. Linearly interpolate the vertex normal over the projected area of the polygon.
3. Apply illumination model at position along each scan line to calculate pixel intensities using interpolated normal vectors.



$$\vec{N} = \frac{y - y_2}{y_1 - y_2}\vec{N}_1 + \frac{y_1 - y}{y_1 - y_1}\vec{N}_2$$

Advantages:

- It displays more realistic highlights on a surface.
- It greatly reduces the Mach band effect.
- It gives more accurate results.

Disadvantages:

- It requires more calculation and greatly increases the cost of shading steeply.

Fast Phong Shading

Fast phong shading approximates the intensity calculations using a Taylor-series expansion and triangular surface patches.

Surface normal at any point (x, y) over a triangle as; $N = Ax + By + C$

A, B, C are determined from the three vertex equations:

$$N_k = Ax_k + By_k + C, \qquad k = 1,2,3$$

With (x_k, y_k) denoting a vertex position.

Omitting the reflectivity and attenuation parameters,

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

We can write this,

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

Where, a, b, c, d are used to represent the various dot product. For e.g.

$$a = \frac{L \cdot A}{|L|}$$

Finally, denominator in eq. (i) can be expressed as a Taylor-series expansion and retain terms up to second degree in x & 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, and so forth.

Comparison of polygon rendering method

Flat shading	Gouraud shading	Phong shading
Computes illumination once per polygon and apply it to whole polygon.	Computes illumination at vertices and interpolates.	Applies illumination at every point of polygon surface.
Creates discontinuous in color.	Interpolates colors along edges and scan line.	Interpolates normal instead of colors.
Problems of Mach bands.	Handles Mach bands problem found in flat shading.	Removes Mach bands completely.
Low cost.	Not so expensive.	More expensive than gouraud shading.
Requires very less processing and is fast in time.	Required moderate processing time.	Requires complex processing and is slower but is more efficient as compared to other shading methods.
Lighting equation used once per polygon.	Lighting equation used at each vertex.	Lighting equation used at each pixel.

References

- **Donald Hearne and M.Pauline Baker**, "Computer Graphics, C Versions." Prentice Hall