illumination and Surface shading

To produce realistic display of a scene two steps are required -

1. Generate perspective projection of objects
2. Applying natural lighting effects to the visible surfaces.

Therefore to achieve a greater realism it is important that various objects in the scene must be coloured appropriately. So This chapter deals with Two Important aspects of lighting & shading.

1. illumination Models → called as lighting model or shading Model that is used to calculate the intensity of light that we should see at given point on surface of an object.

2. Surface rendering algorithm → called surface shading algorithm that uses the intensity calculation from an illumination model to determine light intensity for all the projected pixel positions for various surfaces in a scene.

Surface rendering can be accomplished by interpolating intensities across the surfaces from small set of illumination model calculation.

Surface ↑ Rendering Methods → ① It is Method of interpolating the intensity values across the surface of an object so as to determine the colour of pixels of Surface using illumin Model.

② It determine how a given intensity of incident light is to be used to illuminate a pixel or point.

① Constant Intensity Method → ① flat shading / faceted shading

② It calculates single intensity value for each polygon on the surface. All the points over the surface of polygon are then displayed with same intensity value.

③ It must assume following assumption →

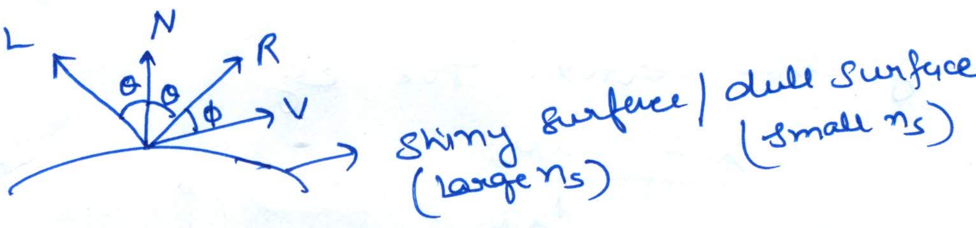
ⓐ The object is a polyhedron and is not an approximation of an object with a curved surface.

ⓑ All light sources illuminating the object are sufficiently far from the surface so that N.L are constant over the surface

ⓒ The viewing position is sufficiently far from the surface so that V.R is constant over the surface.

even if all condn are not true, we can still reasonably approximationat surface rendering effects.

Specular Reflection & Phong Model



① We use L to represent the unit vector directed toward the point light source

② R to represent unit vector in direction of ideal specular Reflection

③ V as the unit vector pointing to the viewer from Surface position.

$$I_{spec} = W(\theta)\, I_l \cos^{n_s}\phi$$

I_l → intensity of light

n_s → specular reflection Parameter

Gouraud Shading → ① developed by Henri Gouraud

② Known as intensity interpolation shading

③ This Method evaluates the illumination formula at the vertices of the polygon mesh. It then renders to each polygon surface by linearly interpolating intensity across the surface.

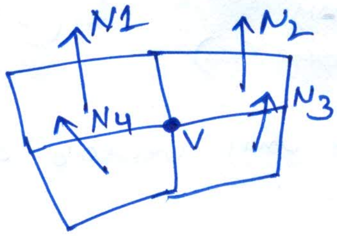
④ It perform the following steps.

ⓐ Determine the avg unit normal vector at each polygon vertex

ⓑ Apply an illumination model to each vertex to calculate the vertex intensity

ⓒ Linearly interpolate the vertex intensities over the surface of polygon.

⑤ At each polygon vertex, we obtain a normal vector by averaging the surface normals of all polygons sharing that vertex. For eg

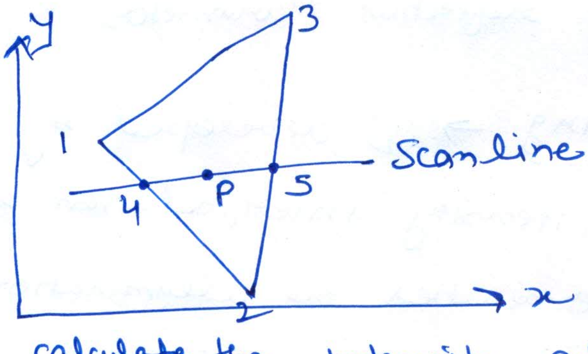


for vertex V, we obtain the unit normal vertex as

$$N_v = \frac{N_1 + N_2 + N_3 + N_4}{|N|} = \frac{\sum_{k=1}^{n} N_k}{\left|\sum_{k=1}^{n} N_k\right|}$$

Once we have the vertex normals, we can determine the intensity at vertices from lighting Model.

⑥ for eg →



① firstly we calculate the intensity at vertex V_1, V_2 & V_3

② Intensity at point 4 is linearly interpolated from the intensities at vertexes 1 & 2. llly for point 5. So point P is then assigned an intensity value that is linearly ~~ploted~~ interpolated from intensities at position 4 & 5.

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

— It removes the intensity discontinuities associated with the constand shading Model. but it has some other deficiencies.

Phong shading → (1) developed by Phong-Bui-Tuong.

(2) normal vector interpolation shading.

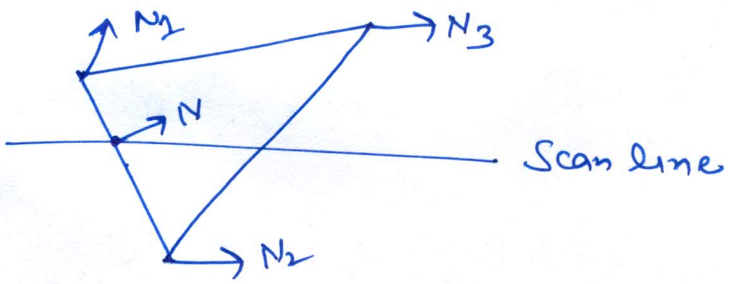
(3) Polygon surfaces is rendered using following steps

(a) Determine the avg unit normal vector at each polygon vertex

(b) Linearly interpolate the vertex normals over the surface of the polygon.

(c) Apply an illumination Model along each scan line to calculate projected pixel intensities for surface points.

(4) Interpolation of surface normals along a polygon edge b/w two vertices is shown as



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

Raytracing, Fractals, Antialiasing, B-spline curve,
Bezier-curve, normalised cordinates, morphing, Appl^n of CG,
Translation is additive in nature, Scaling is commutative,