UNIT 3 POLYGON RENDERING AND RAY TRACING METHODS

Structure **Page Nos.**

3.1 INTRODUCTION

In unit 2 we had discussed some methods for visible-surface detection, but in order to generate visibility the presence of light is one of the basic requirements. It is obvious that without light and the interaction of light with objects, we would never see anything at all. A study of the properties of light and how light interacts with the surfaces of objects is hence vital in producing realistic images in Computer Graphics. So before considering the production of images to be used in animation or any other application it is worth studying some of the basic properties of light and colour and also to introduce the modeling of the interaction of light with surfaces in Computer Graphics because attainment of realism is one of the basic motives of computer graphics and without considering the effect of light the same cannot be achieved. From Principle Physics we can derive models, called "illumination models", of how light reflects from surfaces and produces what we perceive as color. In general, light leaves some light source, e.g., a lamp or the sun, and is reflected from many surfaces and then finally reflected to our eyes, or through an image plane of a camera. In the overall process of reflection, scattering from the objects in the path of light rays there is always production of shadows and shades with varying levels of intensities; this concept of shading is very important in computer graphics because it also contributes to the realism of the scene under preparation. Ray tracing is one of the exercises performed to attain the realism in a scene. In simple terms Ray Tracing is a global illumination based rendering method used for producing views of a virtual 3-dimensional scene on a computer. Ray tracing is closely allied to, and is an extension of, ray casting, a common hidden-surface removal method. It tries to mimic actual physical effects associated with the propagation of light. Ray tracing handles shadows, multiple Specular reflections, and texture mapping in a very easy straightforward manner. In this unit we have a section dedicated to Ray tracing where we intend to inform you how the basic ray tracing algorithm works. We will take a simple approach for the explanation of the concept, avoiding the mathematical perspective which is traditionally used on the subject. It is intended primarily to inform the curious, rather than to teach the ambitious.

3.2 OBJECTIVES

After going through this unit, you should be able to:

- describe types of light sources and their effects;
- discuss Illumination model and different reflections covered in this model;
- discuss the concept of shading and its types, and
- describe the concept of Ray tracing and algorithms used.

3.3 ILLUMINATION MODEL

Conceptually illumination is exposure of an object to the light, which contributes to light reflected from an object to our eyes and this phenomenon in turn determines the color perceived by an object. Thus, if white light is incident on an object then if that object absorbs green and blue light then we shall perceive it as being red. The colour of the light incident on the surface will determine the colour perceived by the viewer, for **example**, if you see red rose in blue light then it will appear black because all blue rays are absorbed by the object and nothing is reflected so it appears black. Similarly, it is the reflectance of the object surface that determines that an object will appear dull or shining; if the object absorbs a high percentage of the light incident on it then it will appear dull whereas if it reflects a large percentage of the light incident on it then it will appear glossy or shiny. For example, if green light were to shine on a red surface then the surface would be perceived as black because a red surface absorbs green and blue.

Thus, to produce realistic computer-generated images of solid opaque objects the various interactions of light with a surface have to be accounted for, in some form of reflected light and for this the Illumination Model is the gift to Computer Graphics from Physics, which will us help to achieve realism in any graphic scene. An illumination model is also called lighting model and sometimes referred to as shading model, which is used to calculate the intensity of the light that is reflected at a given point on surface of an object. Illumination models can be classified as:

Local illumination model: Where only light that is directly reflected from a light source via a surface to our eyes is considered. No account is taken of any light that is incident on the surface after multiple reflections between other surfaces. This is the type of illumination model that is used in most scan-line rendering pipelines. That is the contribution from the light that goes directly from the light source and is reflected from the surface is called a “local illumination model”. So, for a local illumination model, the shading of any surface is independent of the shading of all other surfaces. The scan-line rendering system uses the local illumination model.

Global illumination model: Global illumination model adds to the local model the light that is reflected from other surfaces to the current surface. A global illumination model is more comprehensive, more physically correct, and produces more realistic images. It is also more computationally expensive. In a Global Illumination Model the reflection of light from a surface is modeled as in the local model with the addition of light incident on the surface after multiple reflections between other surfaces. Although the model is computationally more intensive than a local model but attainment of realism through this model is quite possible. The two major types of graphics systems that use global illumination models are Radiosity and Ray tracing.

Radiosity and Ray tracing (The difference in the simulation is the starting point: Ray tracing follows all rays from the eye of the viewer back to the light sources. Radiosity simulates the diffuse propagation of light starting at the light sources). They produce more realistic images but are more computationally intensive than scan-line rendering systems which use local illumination model.

Ray tracing: Ray tracing follows all rays from the eye of the viewer back to the light sources. This method is very good at simulating specular reflections and transparency, since the rays that are traced through the scenes can be easily bounced at mirrors and refracted by transparent objects. We will discuss these concepts Reflection/ Refraction/ transparency when we reach the section of ray-tracking.

Radiosity: Radiosity simulates the diffuse propagation of light starting at the light sources. Since global illumination is a very difficult problem and with a standard ray tracing algorithm, this is a very time consuming task, as a huge number of rays have to be shot. For this reason, the radiosity method was invented. The main idea of the method is to store illumination values on the surfaces of the objects, as the light is propagated starting at the light sources.

Deterministic **radiosity algorithms** were used for radiosity for quite some time, but they are too slow for calculating global illumination for very complex scenes. For this reason, stochastic methods were invented, that simulate the photon propagation using a Monte Carlo type algorithm.

Note: **An illumination model** is also called lighting model and sometimes referred to as shading model, which is used to calculate the intensity of the light that is reflected at a given point on the surface of an object, whereas the **Surface rendering algorithm** uses the intensity calculations from an illumination model to determine the light intensity for all projected pixels positions for the various surfaces in the scene.

From the above discussion we have realised that it’s the types of light source that contributes a lot towards the attainment of realism in any computer graphics scene.

So, let us discuss the types of light sources. The light sources can not only be natural like light from Sun or Moon or Stars but it could be man-made devices like bulb or tube etc., or a highly polished surface. The light sources are referred as **Luminous objects** which are the objects that emit radiant energy and they can be of both types **light emitting source** (which could be of any type point /diffuse/distributed objects emitting radiant energy) and a **light reflecting source** (Reflecting surfaces are sometimes referred to as light reflecting sources, i.e., any polished surface capable of reflecting, a considerable amount of light rays).

Note: When we view an opaque non-luminous object, we see reflected light from one surface of the object. The total reflected light is the sum of each contribution from light sources and other reflecting surfaces in the scene. Thus a surfaces that is not directly exposed to a light source may still be visible if nearby objects are illuminated.

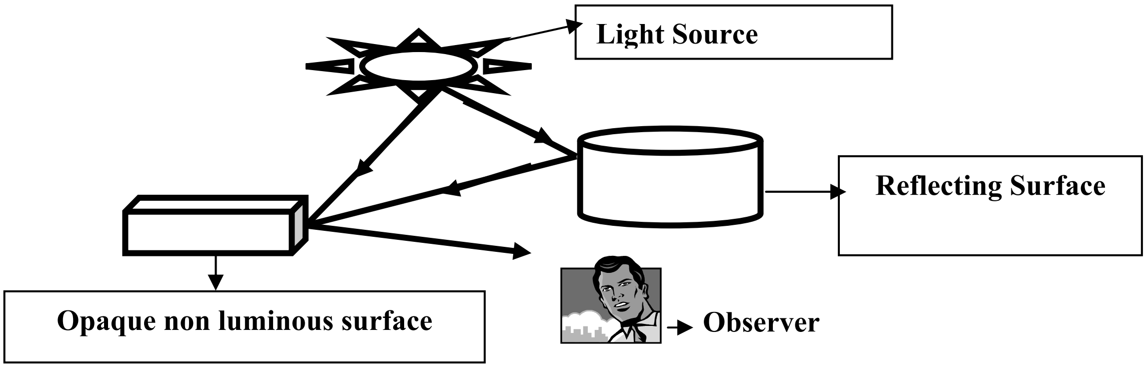


Figure 1

From *Figure 1* we can conclude that the expression given below holds good in real life situations

Light viewed from opaque non-luminous surface = Light from sources + Light from Other Surfaces

Since light sources are quite dominant which are required to establish realism in any graphic scene. Further, there are a variety of light sources, so we need to classify them.

Sources of Light can be classified as:
(a) Point source (b) Parallel Source (c) Distributed Source

Classification of Light Sources

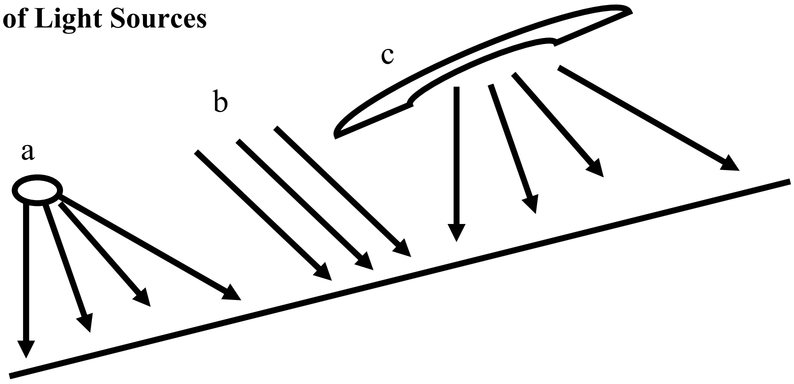


Figure 2: a-Point source; b-Parallel Source; c- Distributed Source

a) **Point source:** It is the simplest model for a light emitter. Here rays from source follow radically diverging paths from the source position, e.g., sources such LED's or small bulbs, i.e., these are the light sources in which light rays originate at a point and radially diverge, such type of sources have dimensions quite smaller as compared to the size of object as shown in *Figure 2*, the source *a* is a point source

b) **Parallel source:** It is to be noted that when point source is at an infinite distance then light rays are parallel and acts as **parallel source** as shown in *Figure 2*, the source b is a parallel source.

c) **Distributed light source:** It models nearby sources such as the long fluorescent light are modeled in category of distributed light source. Here all light rays originate at a finite area in space. Shown in *Figure 2*, the source *c* is a distributed light source

Note: When a light is incident on an opaque surface, part of it is reflected and part of it is absorbed. The amount of incident light reflected by a surface depends on the type of material (shiny material reflect more of the incident light and dull surfaces absorb more of the incident light). Thus, from the reflected amount of light we can deduce many properties of the surface under exposure.

Description of any light source by a luminance, the factors considered are:

Light source described by a luminance

1) Each color (r-red, g-green, b-blue) is described separately

2) $I = [Ir\ Ig\ Ib]^T$ (I for intensity- which is the number of photons incident on a surface in specific time duration).

Now, the interaction of light and the surface under exposure contributes to several optical phenomena like reflection, refraction, scattering, dispersion, diffraction, etc.

☞ Check Your Progress 1

1) Differentiate between Luminous and illuminous objects.

   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

2) What will be the colour of a blue rose when it is viewed in red light? Give reasons in support of your answer.

   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

3) If the source of light is very far from the object what type of rays you expect from the source? What will happen to the type of rays if source is quite close to the object?

   ……………………………………………………………………………………
   ……………………………………………………………………………………
   ……………………………………………………………………………………

Let us discuss reflection and its types:

Reflection: It is the phenomenon of bouncing back of light, this phenomenon follows **laws of Reflection** which are:

First Law of Reflection: The Incident ray, the Reflected ray and the Normal all lie on the same plane.

Second Law of Reflection: The angle of Incidence is equal to the angle of Reflection.

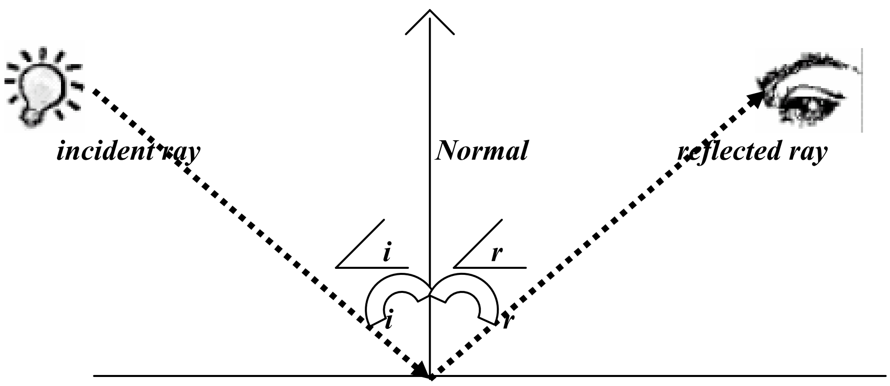


Figure 3: Plane of incidence

Note: Interaction of Light and Surface: A surface has 4 types of interaction with light:

- Diffuse reflection – light is reflected uniformly with no specific direction
- Specular reflection – light is reflected in a specific direction
- Diffuse transmission – light is transmitted uniformly with no preferred direction
- Specular transmission – light is transmitted in a preferred direction.

Types of Reflection: In order to attain realism, this phenomenon of reflection, which occurs due to interaction of light and surface, is needed to be implemented by different ray tracing techniques and other tools. But the usage of tools depends on types of reflection.

- **Ambient Reflection**
- **Diffuse Reflection**
- **Specular Reflection**

Let us discuss different types of reflections.

3.3.1 Ambient Reflection

Whenever we go for the study of light effects, then surroundings play an important role and it is assumed that there exists some light in surroundings falling uniformly on neighbourhood objects. This light in the environment is categorised as Ambient Light (it is non-directional, i.e., it exposes the object uniformly from all directions).

Ambient light is the combination of light reflections from various surfaces to produce uniform illumination which is referred to as Ambient light or Background light. Some features associated with this kind of light are:

- Ambient light has no directional or spatial characteristics,
- The amount of ambient light incident on each object is constant for all surfaces and for all directions.
- The amount of ambient light reflected is dependent on the properties of the surface
- The intensity of ambient light uniform at every point may be different for every surface and color r,g,b

Example: Consider a sphere with a light source above it, thus its lower half will not be illuminated. In practice in a real scene this lower half would be partially illuminated by light that had been reflected from other objects. This effect is approximated in a local illumination model by adding a term to approximate this general light which is 'bouncing' around the scene. This term is called the **ambient reflection** term and is modeled by a constant term. Again the amount of ambient light reflected is dependent on the properties of the surface.It is to be noted that if $I_a \rightarrow$ intensity of ambient light; $K_a \rightarrow$ property of material (**Ambient reflection coefficient k_a , $0 < k_a < 1$**) then resulting reflected light is a constant for each surface independent of viewing direction and spatial orientation of surface.

Say, $I_a \rightarrow$ *Intensity of ambient light.*

$I \rightarrow$ *Intensity of reflected ambient light*

It is assumed that $I_a \neq 0$ ($\because I_a = 0 \Rightarrow$ *These does not exit any light)*

$I \,\alpha\, I_a \quad \Rightarrow I = K_a I_a \quad ; \quad K_a \rightarrow constant \quad ; 0 \leq K_a \leq 1$

$K_a = 0 \Rightarrow$ *object has absorbed the whole incident light.*

$K_a = 1 \Rightarrow$ *object has reflected the whole incident light.*

$0 \leq K_a \leq 1 \Rightarrow$ *object has reflected some and absorbed some light.*

☞ **Check Your Progress 2**

1) How does the value of ambient reflection coefficient deduce the property of material?

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

2) What should be the K_a for a black hole in universe? Give reasons.

……………………………………………………………………………………………

……………………………………………………………………………………………

……………………………………………………………………………………………

3.3.2 Diffuse Reflection

Diffuse reflection is characteristic of light reflected from a dull, non-shiny surface. Objects illuminated solely by diffusely reflected light exhibit an equal light intensity from all viewing directions. That is in Diffuse reflection light incident on the surface is reflected equally in all directions and is attenuated by an amount dependent upon the physical properties of the surface. Since light is reflected equally in all directions the perceived illumination of the surface is not dependent on the position of the observer. Diffuse reflection models the light reflecting properties of matt surfaces, i.e., surfaces that are rough or grainy which tend to scatter the reflected light in all directions. This scattered light is called diffuse reflection.

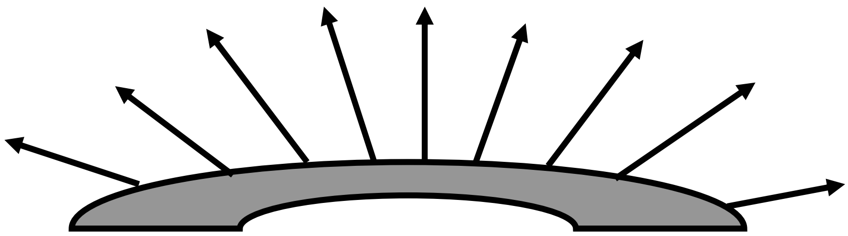


Figure 4: Diffused Reflection From surface

Note:

1) A very rough surface appears equally bright from all viewing directions $\because$ the intensity of reflected light is uniform in all directions thus produce diffuse reflection which are constant over each surface in a scene, independent of the viewing direction.

   The fractional amount of the incident light that is diffusely reflected can be set for each surface with parameter K_d . **K_d →diffuse reflection coefficient or diffuse reflectivity.$0 \le K_d \le 1$(K_d → property of material).**

   $K_d = 1$ for highly reflective surfaces reflecting whole light.
   $K_d = 0$ for surfaces that absorb light fully.

2) **Assumption:** i) The diffuse reflections from the surface are scattered with equal intensity in all directions, independent of viewing direction. Such surfaces are called "ideal diffuse reflectors" or "Lambertion reflectors" $\because$ radiant high energy from any point on the surface is governed by "LAMBERTS COSINE LAW". (i.e., in diffuse reflection ease the intensity of reflected light (**I**) is α **cos** θ and (ii) $K_d \Leftrightarrow$ Ka (generally).

"LAMBERTS COSINE LAW" states that the radiant energy from any small surface area dA in any direction θ relative to the surface normal is proportional to Cos θ.

In case of diffused reflection the source is directional but reflection is uniform.
say,

$I_d \rightarrow$ Intensity of incident diffused light.

Then as per the Lambert's law the intensity of reflected light (I) will be $\alpha \cos\theta$.
Where, θ = Angle between unit direction of incident light vector and unit normal to the surface (or angle of incidence).

/*LAMBERT's LAW * /

$$\boxed{I = K_d\, I_d \cos\theta}$$

$K_d \rightarrow$ diffused reflection coefficient.
$0 \le K_d \le 1$

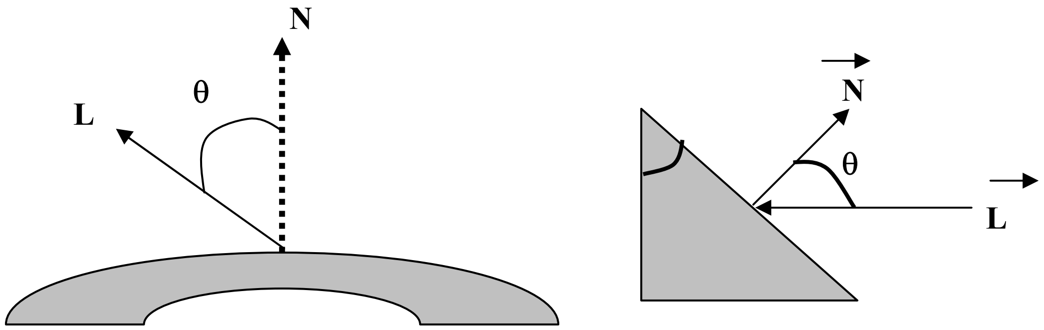


Figure 5

$$\boxed{I = K_d\; I_d\; (\overline{N}.\overline{L})}$$

$I\ \alpha \cos\theta \Rightarrow$ less θ leads to more reflection &
more θ leads to less reflection.

Dot product of N & L vectors $\overline{N}.\,\overline{L} = \underbrace{|\,\overline{N}\,|}_{1}\ \underbrace{|\,\overline{L}\,|}_{1}$ **cos θ** = **cos θ** (∵ $|\,\overline{N}\,|$ & $|\,\overline{L}\,|$ are)

Unit normal to surface **Unit vector in light direction**

3) **Combined effect of ambient and diffused reflection**

Here the resulting intensity I will be the sum total of the intensities in case 8.3.2 & 8.3.3 we get

$$\boxed{I = I_a\, K_a + I_d\, K_d \cos\theta = I_a\, K_a + I_d\, K_d \left(\overline{N}.\overline{L}\right)}$$

Take $K_a = K_d$ (∵ both constant properties of material to which light is incident, for both sources there constant are same).

Example: Consider a shiny surface with diffused reflection coefficient of 0.8 and ambient reflection coefficient of 0.7, the surface has normal in the direction of 2i + 3j +4k; say some light is incident on it from the direction i + j + k such that the ambient and diffused intensities are of order 2 and 3 units. Determine the intensity or reflected light.

Solution: The combined effect of ambient and diffused reflection is given by

$$I = I_a K_a + I_d K_d \cos\theta = I_a K_a + I_d K_d \left(\bar{N}.\bar{L}\right)$$

Using the data given in the equation we get

$$\begin{aligned} I &= 2 * 0.7 + 3 * 0.8 * ((2i + 3j + 4k).(i + j + k)) \\ &= 1.4 + 2.4\,(2 + 3 + 4) \\ &= 1.4 + 9*2.4 \\ &= 23 \end{aligned}$$

3.3.3 Specular Reflection

Specular reflection is when the reflection is stronger in one viewing direction, i.e., there is a bright spot, called a specular highlight. This is readily apparent on shiny surfaces. For an ideal reflector, such as a mirror, the angle of incidence equals the angle of specular reflection, as shown below.

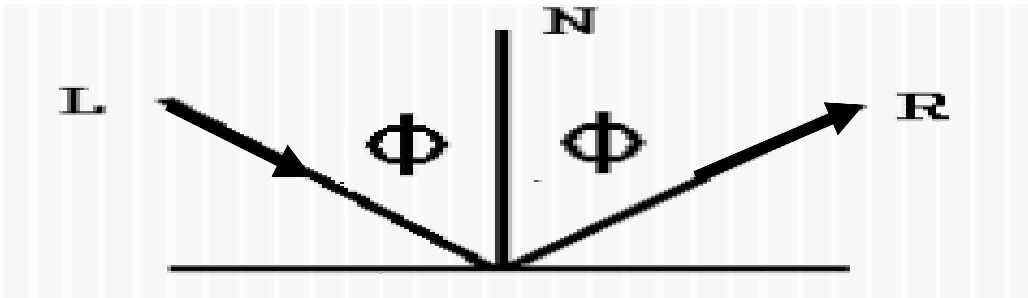


Figure 6

Light is reflected mainly in the direction of the reflected ray and is attenuated by an amount dependent upon the physical properties of the surface. *Since the light reflected from the surface is mainly in the direction of the reflected ray the position of the observer determines the perceived illumination of the surface*. Specular reflection models the light reflecting properties of shiny or mirror-like surfaces.

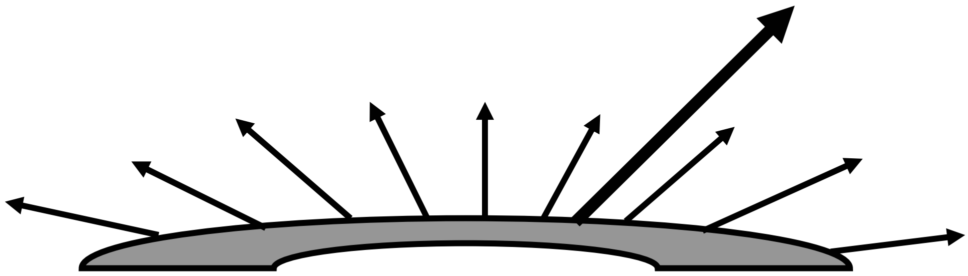


Figure 7

Note: (1) In addition to diffuse reflection, light sources create highlights or bright spots called specular reflection. This highlighting is more pronounced on shiny surfaces than on dull surfaces.

(2) Hence, the local illumination model that is generally used is

illumination = Ambient + Diffuse + Specular

This model of local illumination is usually called the **Phong** specular reflection model.

Let us discuss the concept of specular reflection in a more practical way. Consider the *Figure 9*. Here if **R** is the direction of specular reflection and **V** is the direction of the viewer (located at the View Reference Point or **VRP**), then for an ideal reflector the specular reflection is visible only when **V** and **R** coincide. For real objects (not perfect reflectors) the specular reflectance can be seen even if **V** and **R** don't coincide, i.e., it is visible over range of values (or a cone of values). The shinier the surface, the smaller the f(α) range for specular visibility. So a specular reflectance model must have maximum intensity at **R**, with an intensity which decreases as f(α).

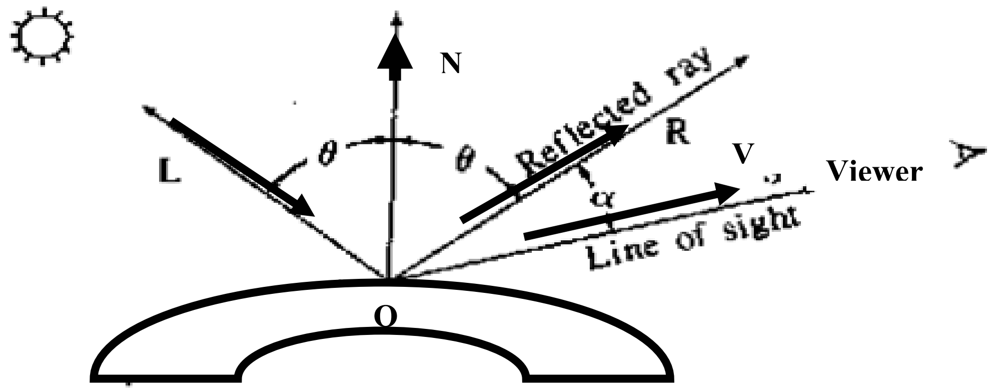


Figure 8

From the above discussion we conclude that Specular reflection is the result of total or near total reflection of the light in a concentrated region around the specular reflection angle (α and the description of other variables shown in *Figure 8* are

$\overline{N}$ → Unit normal surface vector.
$\overline{R}$ → Unit vector in the direction of ideal specular reflection
$\overline{L}$ → Unit vector in the direction of pt. Light source
$\overline{V}$ → Unit vector pointing the viewer.
α → viewing angle relative to $\overline{R}$.

Note:

- At α = 0 viewer will see light of more intensity.
- In case of ideal reflection (perfect mirror) incident light is reflected only in specular reflection direction.
- Objects other than ideal reflection exhibit specular reflection over a finite range of viewing positions around $\overline{R}$ (shiny surfaces have narrow specular reflection range and dull surfaces have wide range).

☞ Check Your Progress 3

1) What will be the change in viewing angle of reflection if the surface under exposure of light is transforming from imperfect reflector to a perfect one?

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

2) If no variation in the intensity of reflection light is observed in any direction, then what can you say about the smoothness of the surface? Also specify what type of reflection you expect from such surface.

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

3) Discuss the law that forms the basis of Lambertion reflections?

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

Phong Model / Phong Specular Reflection Model

This is an empirical model, which is not based on physics, but physical observation. Phong observed that for very shiny surfaces the specular highlight was small and the intensity fell off rapidly, while for duller surfaces it was larger and fell off more slowly. He decided to let the reflected intensity be a function of $(\cos \alpha)^n$ with *n >= 200* for a shiny surface and n small for a dull surface. For a perfect reflector n equals infinity, and for a piece of cardboard *n* equals 0 or 1. In the diagram below we can see how the function $(\cos \alpha)^n$ behaves for different values of *n*. This empirical model for calculating the specular reflection range was developed by Phong and hence called PHONG MODEL/ PHONG SPECULAR REFLECTION MODEL. Which says that, the intensity of specular reflection is proportional to $\cos^n\alpha$ (α lies between 0° & 90°) so $\cos \alpha$ varies from 1 to 0).where '*n*' is *specular reflection parameter* dependent on the type of surface.

Notice that the Phong illumination equation is simply the Lambert illumination equation with an additional summand to account for specular reflection and ambient reflection.

Intensity of specular reflection depends on material properties of surface and the angle of incidence and the value of *specular reflection parameter* '**n**' is determined by the type of surface, we want to display.

- shiny surfaces are modeled with larger values of *n* (100 or more (say))
- dull surfaces are modeled with smaller values of *n* (down to 1)
- for perfect reflection *n* is ∞.
- for very dull surfaces (eq chalk etc) *n* is assigned value near to 1.
- In the diagram below we can see how the function $(\cos \alpha)^n$ behaves for different values of n.

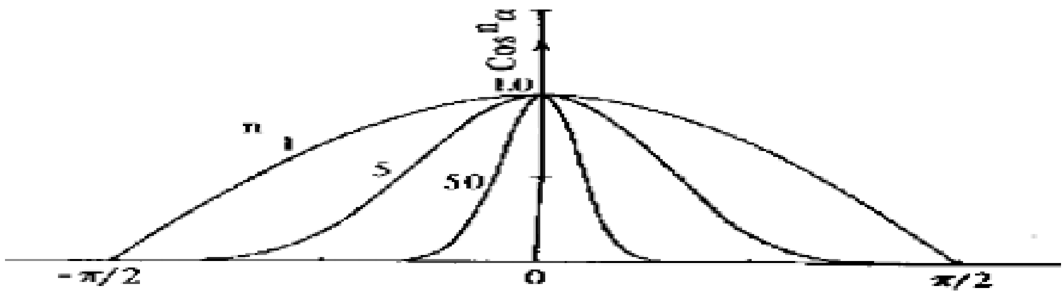


Figure 9: The plot of $\cos^n \alpha$ with α

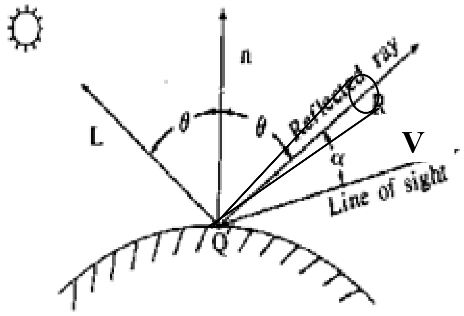


Figure 10: Shiny surface(large n)

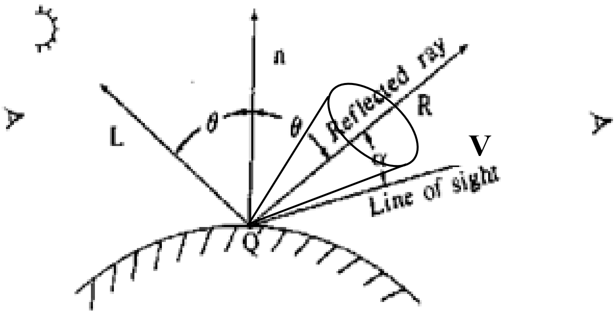
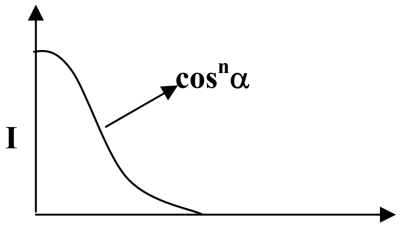
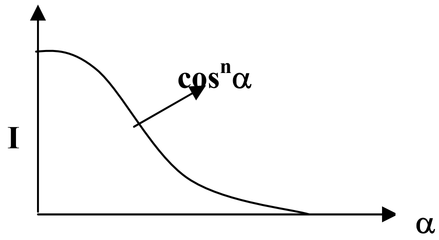


Figure 11: Dull surface(Small n)

As per the phong model variation of Intensity (I) with α (because $I \propto \cos^n \alpha$) is



i) for shiny surface (n > 1)



ii) Dull surface (n ~= 1)

As per the Phong specular reflection,

$$\boxed{I \text{ or } I_{spec.} = I_s K_s \cos^n\alpha}$$

where, $I_s \rightarrow$ intensity of source
I OR $I_{spec} \rightarrow$ intensity of specular reflected light
$K_s \rightarrow$ Specular reflection coefficient

resulting intensity in the case when all ambient/diffuse/ specular reflection occurs is

$$\boxed{I = I_a K_a + I_d K_d \cos\theta + I_s K_s \cos^n \alpha}$$

Now, $\cos^n \alpha = (\overline{R}.\overline{V})^n$ $\qquad (\because \cos\alpha = \overline{R}.\overline{V})$.

Where,

$\overline{R} \rightarrow$ Unit vector in specular reflection direction
$\overline{V} \rightarrow$ Unit vector pointing the viewer

i.e. $|\overline{V}| = 1$; $\quad |\overline{R}| = 1$.

$\overline{R}.\overline{V} = |\overline{R}|\ |\overline{V}| \cos\alpha = 1.1. \cos\alpha = \cos\alpha$.

($\Rightarrow$ cos α = $\overline{R}\,.\,\overline{V}$)

Example: *Calculate* $\overline{R}.\overline{V}$ *using* $\overline{N}, \overline{L}$ & $\overline{V}$ *where the variables have their respective meanings* $\overline{N}$ $\rightarrow$ Unit normal surface vector.

$\overline{R} \rightarrow$ Unit vector in the direction of ideal specular reflection
$\overline{L} \rightarrow$ Unit vector in the direction of pt. Light source
$\overline{V} \rightarrow$ Unit vector pointing the viewer.

Solution: Consider the figure given below

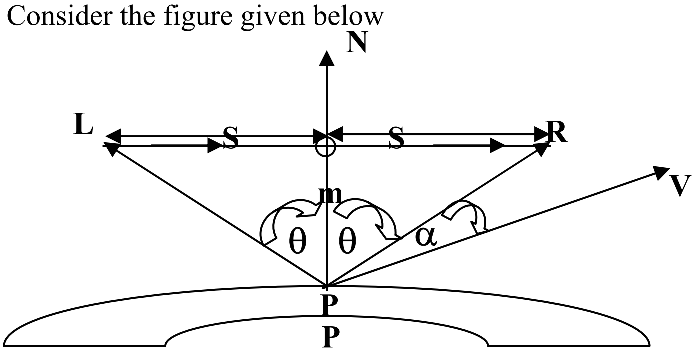


Figure 12

$|\overline{Pm}| = |\overline{L}\ \cos\theta| = \cos\theta$ ------------------(1) $\quad$ ($|\overline{L}| = 1$ (∵ $\overline{L}$ is unit vector))

$\overline{Pm} = |\overline{Pm}|.\ \hat{Pm} = |\overline{Pm}|\ \overline{N}$ ($P\hat{m}$ has direction of normal $\overline{N}$)

$\Rightarrow \overline{Pm} = \overline{N}\ \cos\theta$ ----------(2)

By figure vectors S and R can be found to be

$\overline{S} = \overline{Pm} - \overline{L}$ ---------------------------- (3)

$\overline{R} = \overline{Pm} + \overline{S}$ -----------------------------(4)

Use (3) in (4) we get

$\overline{R} = 2\,\overline{Pm} - \overline{L}$ --------------------------------(5)

Use (2) in (5) we get

$\overline{R} = 2\,(\overline{N}\ \cos\theta) - \overline{L}$

$\Rightarrow \overline{R}\ .\ \overline{V} = [\,2\,(\overline{N}(\cos\theta) - \overline{L}\,]\ .\ \overline{V}.$

$\overline{R}\ .\ \overline{V} = [2\,(\overline{N}(\ \overline{N}\ .\ \overline{L}\,) - \overline{L}\,]\ .\ \overline{V}$ $\quad$ (∵ $\cos\theta = \overline{N}\ .\ \overline{L}$).

3.4 SHADING

When an object is under the exposure of light, then the rays of light are distributed over the surface and the distribution of intensity pattern depends very much on the shape of object. Now to represent such 3D scene on computer we need to do rendering, i.e., transforming the 3D image into a 2D image, and because of this rendering there is a possibility of loss of information like depth, height, etc., of an object in the scene. So to preserve this takes different Illumination models into consideration, for preserving the information embedded in 3D scene & let it not be lost while transforming it in to 2D scene.

So let us study in some detail the type of shading and rendering techniques.

Polygon –Rendering Methods

Here we will consider the application of an illumination model to perform the rendering of standard graphics objects, i.e., those formed with polygonal surfaces. Scan line algorithm typically applies lighting model to perform rendering of polygon surfaces, by using any one of the two ways:

1) Each polygon can be rendered with a single intensity.
2) The intensity can be obtained at each point of the surface using an interpolation scheme.

The result of two ways leads to 3 types of shading:

(a) **Constant intensity shading OR Flat shading**

In this method single intensity is calculated for each polygon surface i.e., all points which lie on the surface of the polygon are displayed with the same intensity value. This constant shading is useful for quickly displaying the general appearance of a curved surface, but this shading does not convey explicit information about the curved surface.

(b) **Gourand shading OR Intensity interpolation scheme**
We will discuss this scheme in successive section 3.4.1.

(c) **Phong shading OR Normal vector interpolation shading.**
We will discuss this scheme in successive section 3.4.2.

3.4.1 Gourand shading OR Intensity interpolation scheme

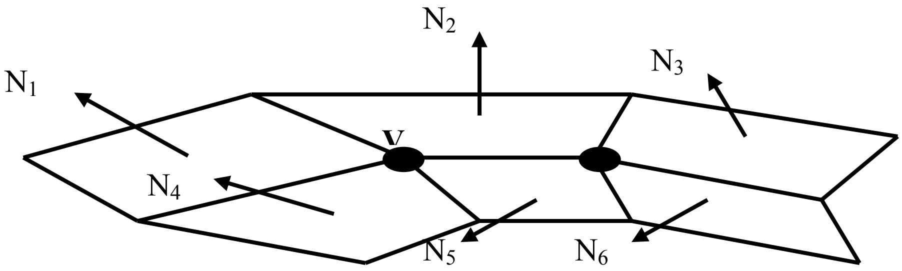


Figure 13

Here polygon is rendered by linearly interpolating intensity values across the surface. Intensity values for each polygon are matched with the values of adjacent polygons along the common edges, thus eliminating the intensity discontinuities that can occur in flat shading.

Calculations to be performed for each polygon surface rendered with Gourand shading:

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

i) *To determine average unit normal vector at each polygon vertex:*

At each polygon vertex (as shown by point V in the figure), the normal vector is obtained by averaging the surface normal of all polygons sharing that vertex. Thus, for any vertex *V* the unit vertex normal will be given by $\overrightarrow{N_v}$

$$\boxed{\overline{N_V} = \frac{\sum_{=1}^{n} N_k}{\left| \sum_{k=1} N_k \right|}}$$

$K \rightarrow 1$ to n are the surfaces in contact with the vertex *v*.

ii) *How to use illumination model to calculate vertex intensity:*

For this we interpolate intensities along the polygon edges, for each scan line the intensity at the intersection of the scan line with a polygon edge is linearly interpolated from intensities at the edge end points, i.e., by using parametric equation of line discussed in Unit 2 we can find intensity at point 4, i.e., I_4; during the evaluation of I_4 intensities at point. 1 & 2 i.e., I_1 & I_2 are used as the extreme intensities. So I_4 is linearly interpolated from intensities at the edge end points I_1 & I_2 (Refer to the Figure given below).

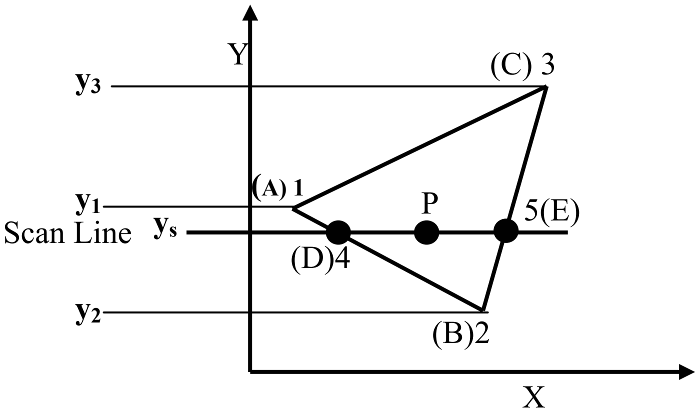


Figure 14

(iii) Linearly interpolate

$$\begin{cases} I_4 = I_1 + t(I_2 - I_1) & \text{where } t = \dfrac{|y_1 - y_s|}{|y_1 - y_2|} \\ I_D = I_A + t(I_B - I_A) & \text{where, } t = \dfrac{|AD|}{|AB|} \\ \text{similarly } I_E = I_C + t(I_B - I_C) & \text{where } t = \dfrac{|CE|}{|CB|} \\ \& \; I_P = I_D + t\;(I_E - I_D) & \text{where } t = \dfrac{|EP|}{|DE|} \end{cases}$$

where $I_p \rightarrow$ Intensity of points over the surface of polygon i.e., in Gourand shading the intensity of point - 4 is linearly interpolated from intensity at vertices 1 and 2, similarly of point 5 too is interpolated from intensity at vertices 3 and 2. Intensity of points *P* is linearly interpolated from intensity at point 4 and 5.

Advantages of Gourand Shading: It removes the intensity discontinuities associated with the constant shading model.

Deficiencies: Linear intensity interpolation can cause bright and dark streaks called Mach bands to appear on the surface, these mach bands can be removed by using Phong shading or by dividing the surface into greater number of polygon faces.

Note: In Gourand Shading because of the consideration of average normal, the intensity is uniform across the edge between two vertices.

3.4.2 Phong shading OR Normal Vector Interpolation Shading

In Gouraud shading we were doing direct interpolation of intensities but a more accurate method for rendering a polygon surface is to interpolate normal vectors and then apply illumination model to each surface. This accurate method was given by Phong and it leads to Phong shading on Normal vector interpolation shading.

Calculations involved with Phong Shading:

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

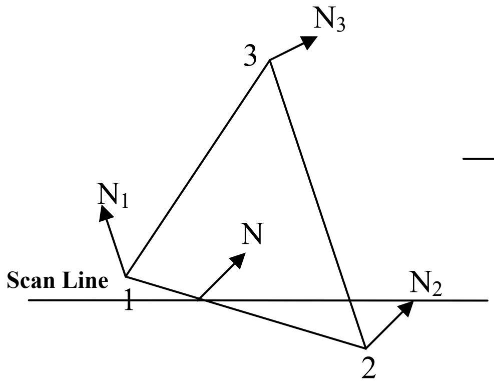


Figure 15

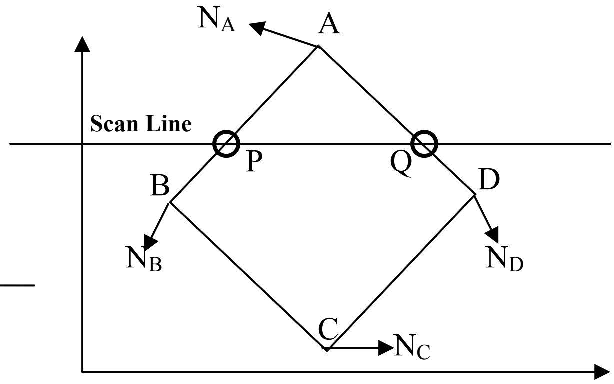


Figure 16

Interpolation of surface normals along the polygonedge between two vertices is shown above in *Figure 15*. The normal vector *N* for the scan line intersection point along the edge between vertices 1 and 2 can be obtained by vertically interpolating between edge end points normals. Then incremental methods are used to evaluate normals between scan lines and along each individual scan line. At each pixel position along a scan line , the illumination model is applied to determine the surface intensity at that point

$$N= [((y-y_2) / (y_1-y_2)) N_1] + [((y-y_2) / (y_1-y_2)) N_2]$$

In *Figure 15* above, say, $\overline{N}$ is surface normal to be interpolated along polygon edge 1– 2 having vertices 1 & 2. Such that $\overline{N}_1$ & $\overline{N}_2$ are normal at the vertices. Thus, by using the parametric equation across the edge 1 – 2 we can determine the value of the normal *N*, which will be given by

$$\overline{N} = \overline{N}_1 + t\ (\overline{N}_2 - \overline{N}_1)$$

Similarly, in *Figure 16* we can find $\overline{N}_p$ and $\overline{N}_q$ which are the normal at point (*P* and *Q*) through which the scan line passes,

$$\boxed{\overline{N}_P = \overline{N}_A + t\ (\overline{N}_B - \overline{N}_A)} \quad \text{where, } t = \frac{|AP|}{|AB|}$$

Now we use $\overline{N}_P$ to find cos θ where , θ is the angle between Normal vector and direction of light represented by vector L(refer to Phong model).

$$\cos\theta = \overline{N}.\ \overline{L} \quad \text{and} \quad \cos^n \alpha = (\overline{R}.\overline{V})^n = [(2\overline{N}(\overline{N}.\ \overline{L}) - \overline{L}).\ V]^n$$

Now using $\cos^n \alpha$, $\cos\theta$ in

$$\boxed{I = I_a K_a + I_d K_d \cos\theta + I_s K_s \cos^n \alpha}$$

/* similarly we can find intensity of points lying inside the surface */

This N_p will be used to find intensity value i.e., I_P. at points P_o in the object whose projection is P, by using the intensity calculation formula which we had used for the determination of intensities in diffused and specular reflection.

Merit

So by finding intensities at different points across the edge we find that intensity is varying across the edge between two vertex points and is not uniform as in Gouraud Shading, giving much better effect. So we can say that the normal vector interpolation (Phong shading) is superior to intensity interpolation technique (Gourand Shading) because it greatly reduces the mach bands.

Demerit: Requires lot of calculations to find intensity at a point, thus increases the cost of shading in any implimentation.

Problem with Interpolated Shading

There are some more shading models which intermediate in complexity between Gouraud and Phong shading, involving the liner interpolation of the dot products used in the illumination models. As in Phong shading, the illumination model is evaluated at each pixel, but the interpolated dot products are used to avoid the expense of computing and normalizing any of the direction vectors. This model can produce more satisfactory effects than Gouraud shading when used with specular-reflection illumination models, since the specular term is calculated separately and has power-law, rather than linear, falloff. As in Gouraud shading, however, highlights are missed if they do not fall at a vertex, since no intensity value computed for a set of interpolated dot products can exceed those computed for the set of dot products at either end of the span.

There are many problems common to all these interpolated-shading models, several of which we listed here.

Polygonal silhouette: No matter how good an approximation an interpolated shading model offers to the actual shading of a curved surface, the silhouette edge of the mesh is still clearly polygonal. We can improve this situation by breaking the surface into a greater number of smaller polygons, but at a corresponding increase in expense.

Perspective distortion. Anomalies are introduced because interpolation is performed after perspective transformation in the 3D screen-coordinate system, rather than in the WC system. For example, linear interpolation causes the shading information to be incremented by a constant amount from one scan line to another along each edge. Consider what happens when vertex 1 is more distant than vertex 2. Perspective foreshortening means that the difference from one scan line to another in the untransformed *z* value along an edge increases in the direction of the farther coordinate. Thus, if $y_s = (y_1 + y_2)$, then $I_s = (I_1 + I_2)/2$, but z_s will not equal $(z_1 + z_2/2$. This problem can also be reduced by using a larger number of smaller polygons. Decreasing the size of the polygons increases the number of points at which the information to be interpolated is sampled, and therefore increases the accuracy of the shading.

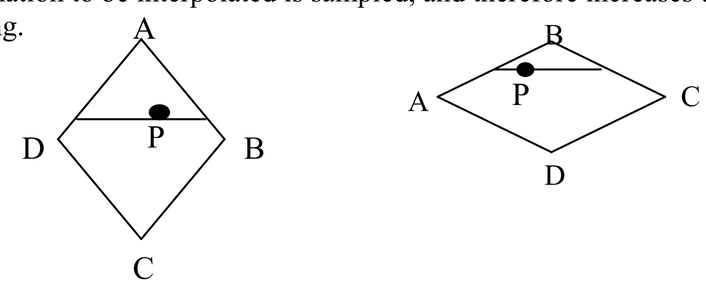


Figure 17: (a) **(b)**

Figure 17 (a) and 17 (b) shows Interpolated values derived for point *P* on the same polygon at different orientations which differ from (a) to (b). *P* interpolates A, B, D in (a) and A, B, C in (b).

Orientation dependence: The results of interpolated-shading models are not independent of the projected polygon's orientation. Since values are interpolated between vertices and across horizontal scan lines, the results may differ when the polygon is rotated (see Figure 17). This effect is particularly obvious when the orientation changes slowly between successive frames of an animation. A similar problem cab also occurs in visible-surface determination when the z value at each point is interpolated from the z values assigned to each vertex. Both problems can be solved by decomposing polygons into triangles. Alternatively, the solution is rotation-independent, but expensive, interpolation methods that solve problem without the need for decomposition.

Problems at shared vertices: Shading discontinuities can occur when two adjacent polygons fail to share a vertex that lies along their common edge. Consider the three polygons of *Figure 17*, in which vertex *C* is shared by the two polygons on the right, but not by the large polygon on the left. The shading information determined directly at *C* for the polygons at the right will typically not be the same as the information interpolated at that point from the values at *A* and *B* for the polygon at the left. As a result, there will be a discontinuity in the shading. The discontinuity can be eliminated by inserting in the polygon on the left an extra vertex that shares *C's* shading information. We can preprocess a static polygonal database in order to eliminate this problem; alternatively, if polygons will be split on the fly (e.g., using the BSP-tree visible-surface algorithm), then extra bookkeeping can be done to introduce a new vertex in an edge that shares an edge that is split.

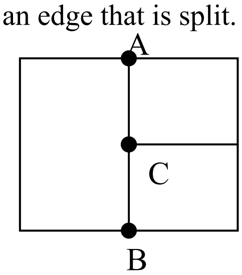


Figure 18: Vertex C is shared by the two polygons on the right, but not by the larger Rectangular polygon on the left.

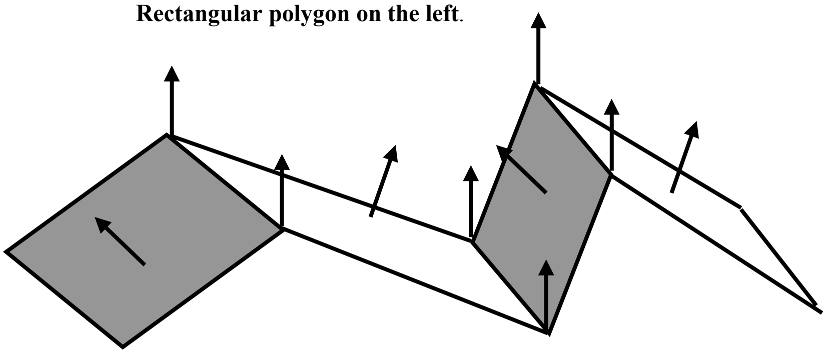


Figure 19: Problems with computing vertex normal. Vertex normal are all parallel.

Unrepresentative vertex normals: Computed vertex normals may not adequately represent the surface's geometry. For example, if we compute vertex normals by averaging the normals of the surfaces sharing a vertex, all of the vertex normals of *Figure 19* will be parallel to one another, resulting in little or no variation in shade if the light source is distant. Subdividing the polygons further before vertex normal computation will solve this problem.

Although these problems have prompted much work on rendering algorithms that handle curved surfaces directly, polygons are sufficiently faster (and easier) to process that they still form the core of most rendering systems.

☞ Check Your Progress 4

1) If specular reflection parameter is small, say 50, than what can you say about the nature of surface? How the nature of surface will change if n starts increasing?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

2) Usage of linear interpolation scheme to obtain intensity at each polygon of surface leads to which type of shading?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

3) What are merits & demerits of Ground shading and Phong shading?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

4) Which shading scheme is best constant shading, Ground shading or Phong Shading? Give reasons to support your answer.

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

5) Compare Constant shading, Gourand shading and Phong Shading.

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

6) Why Phong Shading is better than Gourand shading.

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

7) How Ambient, Diffused and Specular reflection contributes to the resulting intensity of reflected ray of light? Give mathematical expression for the same.

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

8) What do you mean by polygon rendering?

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

3.5 RAY TRACING

Ray tracing is an exercise performed to attain the realism in a scene. In simple terms Ray Tracing is a global illumination based rendering method used for producing views of a virtual 3-dimensional scene on a computer. Ray tracing is closely allied to, and is an extension of, ray-casting, a common hidden-surface removal method. It tries to mimic actual physical effects associated with the propagation of light. Ray tracing handles shadows, multiple specular reflections, and texture mapping in a very easy straight-forward manner. So, the **crux** is "Ray tracing is a method of generating realistic images by computer, in which the paths of individual rays of light are followed from the viewer to their points of origin". Any program that implements this method of ray tracing is ray tracer. One of the prime **advantages** of method of Ray tracing is, it makes use of the actual physics and mathematics behind light. Thus the images produced can be strikingly, life-like, or "photo-realistic".

In this section we will discuss the basic ray-tracing algorithm. It will also describe the concept behind anti-aliasing, a method for improving the realism of an image by smoothing the jagged edges caused by the digital nature of computer displays. This section will not involve any discussions of the more advanced features of today's ray-tracers, such as motion blur, depth of field, penumbras (soft shadows), texture mapping, or radiosity.

So to proceed the journey of ray tracing we will begin with basics like concept of **scene** and describe its basic elements. With this as a foundation, it will then introduce **ray casting**, and then **ray tracing** as an extension of ray casting. Finally, the section will discuss the basic concepts behind **anti-aliasing** as a means of improving the realism of an image, and will then conclude with an overview of **how and where ray tracing is used.**

Scenes

In the context of ray tracing, a scene is a collection of objects and light sources that will be viewed via a camera. Each of these items are arranged in what is called the world, or world space which is an imaginary place with height, width and depth (much like reality). For instance, let's suppose that you wish to create an image of the Planets and their satellites. Then our scene will consist of the planets, their respective satellites, a light source that will act as the sun, and our camera, which is where we are viewing this scene from. Let us discuss some basic terms (objects, light source, world, etc.) involved with the concept of scene in some detail.

Objects could be any state of matter (solid, liquid. gas, plasma). Although ray tracers can only support objects that can have mathematical description (such as cube, spheres, cylinders, planes, cones, etc.), various combinations of these basic objects help in creating more complex objects.

It is important to note that all objects have some kind of texture, which includes the color of the object, as well as any bumpiness, shininess, or design that the designer of the image may wish to use. However, to simplify the discussion of how ray tracing works, we will consider color to be the only texture present on the objects that will be described.

Light Sources are key elements in any ray traced scene, because without them, there would be no rays to trace. Light sources are like objects which may be placed at arbitrary locations in the scene, but in addition to the location a light source has some intensity associated with it, which describes the brightness and color of the light. At any rate, lighting is considered by many to be one of the most important factors in a ray traced image. It is the location and intensity of light source which give liveliness to an image, A picture flooded with too much light might lose any sense of mystery you desired it to have, whereas an image that is too dark will not show enough detail to keep its viewers interested.

Camera represents "eye" or "viewpoint" of observer. In order to describe the basic working of camera we can refer to the working of "pin-hole camera" as shown in the figure below:

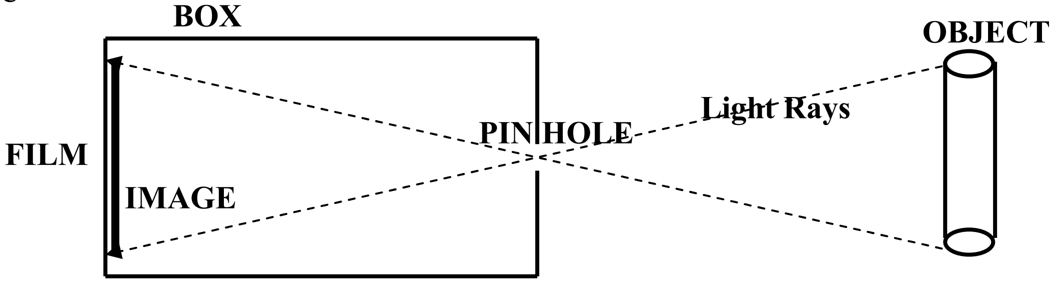


Figure 20

The hole (or "aperture") must be so small that it must prevent light from saturating (and thus overexposing) the film. It allows only a little bit of light into the box at a time. This kind of camera, though simple, is quite effective. It works because light from a given position on the object may come from only one direction and strike only one position on the film. If the hole were any larger, the image would become blurrier as a result of the increased amount of light hitting each spot on the film

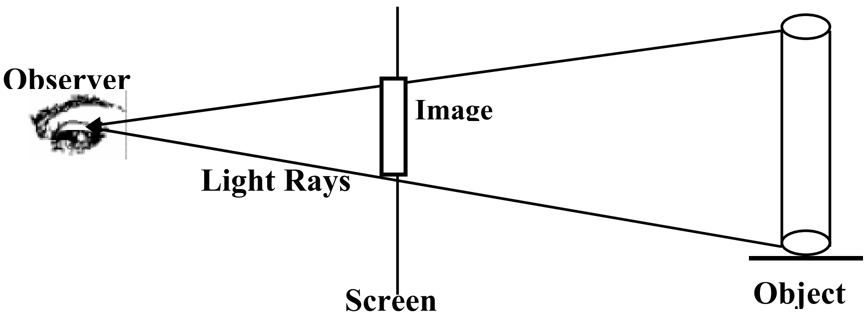


Figure 21

In ray tracing, the camera is much like this, in that it determines where on the "film" (or, in the case of ray tracing, the computer screen) the light rays hit such that a clear realistic rendered image is available. In order to have a better understanding of the topic let us have a discussion on the concept of ray casting also.

Ray Casting

Ray casting is a method in which the surfaces of objects visible to the camera are found by throwing (or casting) rays of light from the viewer into the scene. The idea behind ray casting is to shoot rays from the eye, one per pixel, and find the closest object blocking the path of that ray – think of an image as a screen-door, with each square in the screen being a pixel. This is then the object the eye normally sees through that pixel. Using the material properties and the effect of the lights in the scene, this algorithm can determine the shading of this object. The simplifying assumption is made that if a surface faces a light, the light will reach that surface and not be blocked or in shadow. The shading of the surface is computed using traditional 3D computer graphics shading models. Ray casting is not a synonym for ray tracing, but can be thought of as an abridged, and significantly faster, version of the ray tracing algorithm. Both are image order algorithms used in computer graphics to render three dimensional scenes to two dimensional screens by following rays of light from the eye of the observer to a light source. Although ray tracing is similar to ray casting, it may be better thought of as an extension of ray casting we will discuss this in the next topic under this section.

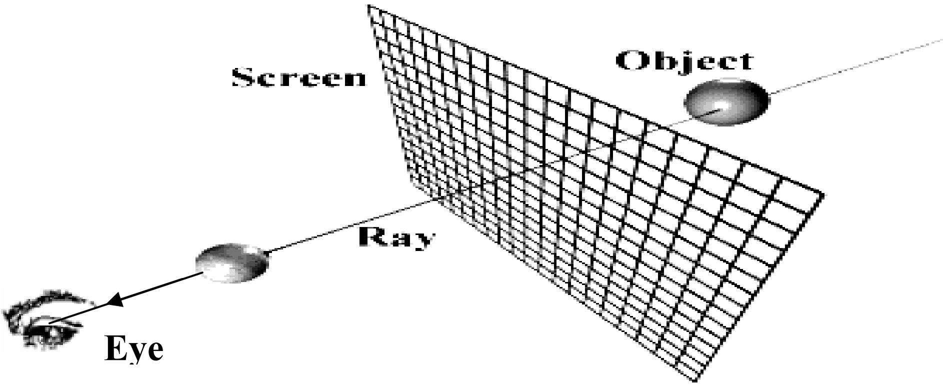


Figure 22

Ray Tracing

"Ray tracing" is a method of following the light from the eye to the light source. Whereas ray casting only concerns itself with finding the visible surfaces of objects, ray tracing takes that a few steps further and actually tries to determine what each visible surface looks like. Although it will cost your processor time spent in calculations you can understand the level of calculations involved in ray tracing by considering this example,

Let's say we are rendering (that is, ray tracing) a scene at a resolution of 320 pixels wide by 240 pixels high, for a total of 76,800 pixels. Let it be of low complexity, with only 20 objects. That means, over the course of creating this picture, the ray tracer will have done 20 intersection tests for each of those 76,800 pixels, for a total of 1,536,000 intersection tests! In fact, most ray tracers spend most of their time calculating these intersections of rays with objects, anywhere from 75 to 95 % of a ray tracer's time is spent with such calculations. Apart from such hectic calculations, there is the good news that there are ways to decrease the number of intersection tests per ray, as well as increase the speed of each intersection test. In addition to this the bad news is that ray tracing complicates things much more than simply ray casting does.

Ray tracing allows you to create several kinds of effects that are very difficult or even impossible to do with other methods. These effects include three items common to every ray tracer: reflection, transparency, and shadows. In the following paragraphs, we will discuss how these effects fit naturally into Ray tracing.

3.5.1 Basic Ray Tracing Algorithm

The Hidden-surface removal is the most complete and most versatile method for display of objects in a realistic fashion. The concept is simply to take one ray at a time, emanating from the viewer's eye (in perspective projection) or from the bundle of parallel lines of sight (in parallel projection) and reaching out to each and every pixel in the viewport, using the laws of optics.

Generally, to avoid wastage of effort by rays starting from sources of light going out of the viewport, the reverse procedure of starting with ray from the viewpoint and traveling to each pixel in the viewport is adopted.

If the ray encounters one or more objects, the algorithm filters out all but the nearest object, or when there is an overlap or a hole, the nearest visible portion of all the objects along the line of sight.

Depending on the nature (attributes) of the surface specified by the user, the following effects are implemented, according to rules of optics.

a) Reflection (according to the angle of incidence and reflectivity of the surface).

b) Refraction (according to the angle of incidence and refraction index).

c) Display of renderings (texture or pattern as specified), or shadows (on the next nearest object or background) involving transparency or opacity, as the case may be.

Figure illustrates some of the general principles of ray tracing.

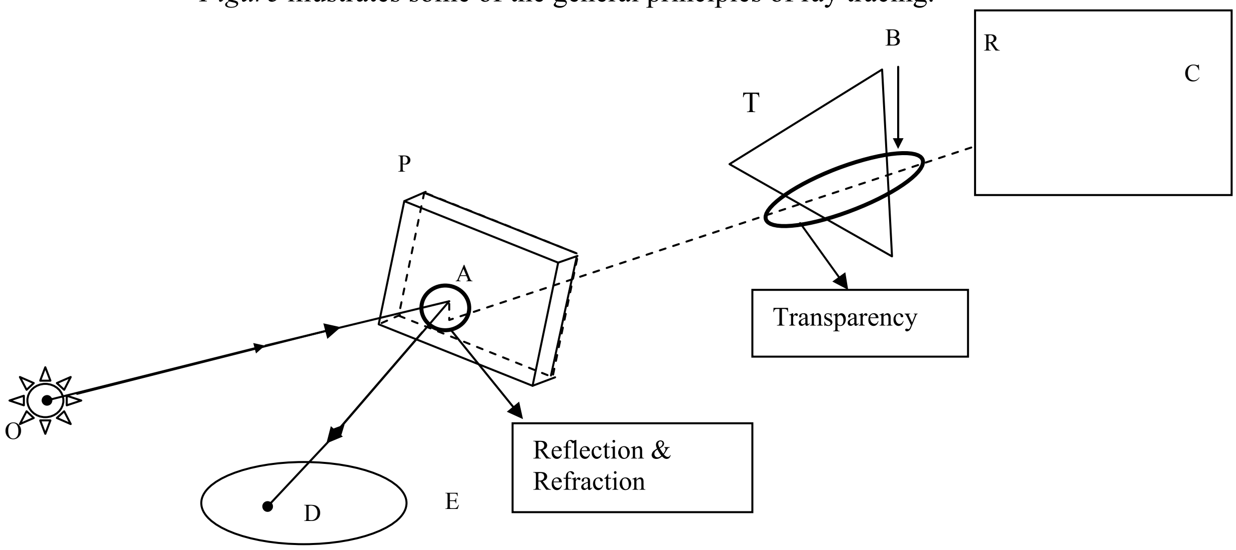


Figure 23

A ray starting at *O* hits the transparent glass plate *P* at an angle at *A*. It gets refracted in the plate (indicated by the kink within the plate thickness). The exiting ray happens to hit the edge of the triangle *T*, and casts a shadow on the opaque rectangular plate *R* at the point *C*. A part of the ray incident on the plate *P* gets reflected at *A* and the reflected ray hits the elliptical object *E* at the point *D*. If *P* is a green glass plate, the exiting ray *AC* will be assigned the appropriate green colour. If *R* or *E* has a textured surface, the corresponding point *C* or *D* will be given the attributes of the surface rendering.

If *O* is a point source of light, the ray *OC* will locate the shadow of the point *B* on the edge of the triangle *T* at the point *C* on the rectangle *R*.

Different locations of light sources may be combined with differing view positions to improve the realism of the scene. The method is general also in the sense that it can apply to curved surfaces as well as to solids made of flat polygonal segments. Because of its versatile and broad applicability, it is a "brute force" method, involving massive computer resources and tremendous computer effort.

Algorithm

Often, the basic ray tracing algorithm is called a "recursive" (obtaining a result in which a given process repeats itself an arbitrary number of times) algorithm. *Infinite recursion* is recursion that never ends. The ray tracing algorithm, too, is recursive, but it is finitely recursive. This is important, because otherwise you would start an image rendering and it would never finish!

The algorithm begins, as in ray casting, by shooting a ray from the eye and through the screen, determining all the objects that intersect the ray, and finding the nearest of those intersections. It then *recurses* (or repeats itself) by shooting more rays from the point of intersection to see what objects are reflected at that point, what objects may be seen through the object at that point, which light sources are directly visible from that point, and so on. These additional rays are often called secondary rays to differentiate them from the original, primary ray. As an analysis of the above discussion we can say that we pay for the increased features of ray tracing by a dramatic increase in time spent with calculations of point of intersections with both the primary rays (as in ray casting) and each secondary and shadow ray. Thus achieving good picture quality, is not an easy task, and it only gets more expensive as you try to achieve more realism in your image. One more concept known as Anti-aliasing is yet to be discussed, which plays a dominant role in achieving the goal of realism.

Anti-aliasing

Anti-aliasing is a method for improving the realism of an image by removing the jagged edges from it. These jagged edges, or "jaggies", appear because a computer monitor has square pixels, and these square pixels are inadequate for displaying lines or curves that are not parallel to the pixels and other reason is low sampling rate of the image information, which in turn leads to these jaggies (quite similar to star casing discussed in previous blocks under DDA algorithm). For better understanding, take the following image of darkened circle:

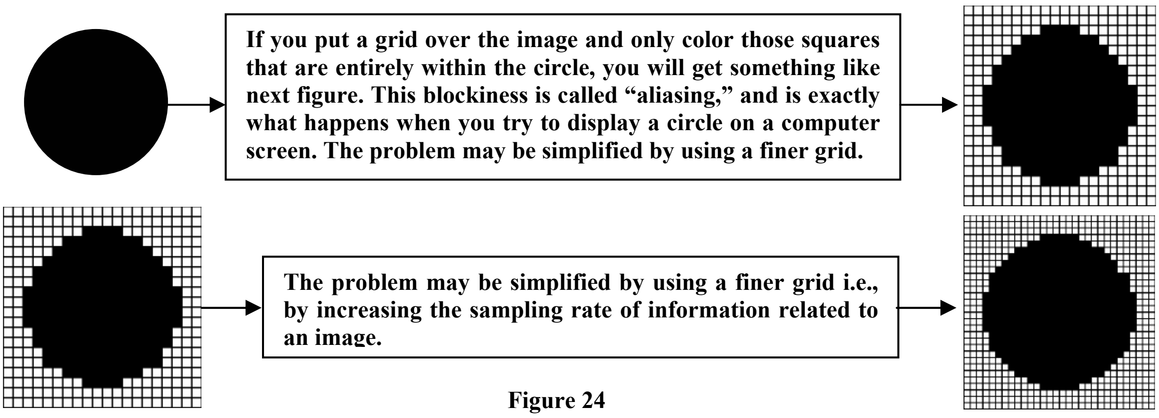
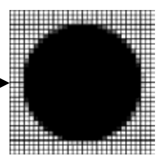


Figure 24

It is not possible to completely eliminate aliasing because computers are digital (discrete) in nature. However, it is possible to minimize aliasing, the solutions used by ray tracers today involve treating each pixel as a finite square area (which, in fact, they are), rather than as a mere point on the screen. Instead the pixel should not be considered as a point or area but should be considered as a sample of image information (higher the sampling is lesser the aliasing is). Now let us discuss how appropriately the sampling can be done - Rays are fired into the scene through the centers of the pixels, and the intensities of adjacent rays are compared. If they differ by some pre-determined amount, more rays are fired into the surfaces of the pixels. The intensities of all the rays shot into a given pixel are then averaged to find a color that better fits what would be expected at that point.

Note: Do not treat a pixel as a square area, as this does not produce correct filtering behaviour, in fact a pixel is not a point, but it is a sample of information to be displayed.

Anti-aliasing, then, helps eliminate jagged edges and to make an image seem more realistic.Continuing the above example, the anti-aliased circle might, then, be represented



Applications of Ray Tracing

So, you might ask, just what practical uses does ray tracing have

- "simulation of real-world phenomena for vision research,
- medical (radiation treatment planning),
- seismic (density calculations along a ray),
- mechanical engineering (interference checking),
- plant design (pipeline interference checking),
- hit-testing in geometric applications, and impact and penetration studies".
- widely used in entertainment.

This topic discussed has just scratched the surface of the topic of ray tracing. In addition to the primitive features of reflection, transparency, and shadows, most ray tracers today support different texture mapping options, focal blur, radiosity, motion blur, and a host of other advanced features. But they are out of the scope of this unit.

☞ Check Your Progress 5

1) Differentiate between Ray tracing and Ray Casting.

…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………

2) How Ray does tracing contribute to achieving realism in computer graphics?

…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………

3) What is the problem of Aliasing? How does the technique of anti-aliasing work to get rid of the problem of aliasing?

…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………
…………………………………………………………………………………………

3.6 SUMMARY

In the unit, we have discussed various sources of light and the type of reflections produced by the object under exposure of these sources. When an object is exposed to light various phenomena occur such as reflection, refraction, etc. Out of them shading is most important as the same helps in achieving the realism in computer graphics. We have also discussed the concept of polygon rendering, and ray tracing which are useful in achieving photo realism of a 3D scene.

3.7 SOLUTIONS / ANSWERS

Check Your Progress 1

1) Luminous objects are independent sources of light e.g., Sun, whereas illuminous objects are those sources of light which depend on luminous sources to produce their half of light, e.g., Moon.

2) It will appear black because nothing of the incident light is reflected back.

3) When the source of light is far off, i.e., at infinite distance (comparatively very large distance) the rays of light become parallel, hence the source behaves as a parallel source.

   But as the source of light appears closer to the object these rays of light start following radially outward path and hence transform to point source of light.

Check Your Progress 2

1) As the coefficient of Ambient reflection (K_a) lies between 0 and 1, the lesser value (value closer to 0) conveys the message that the surface is more absorbing in nature and similarly the surfaces having values of K_a closer to 1, are highly reflective.

2) As Black hole in universe reflects nothing so the value of K_a should be 0.

Check Your Progress 3

1) As the surface under exposure transforms from imperfect reflector to the perfect one, the respective viewing angle of reflection keeps on decreasing because the light reflected will be having more precise direction.

2) Since no variation in intensity of light is observed in any direction, it implies the surface is rough and it which leads to diffused reflection. If the surface is smooth and shiny then reflection should be in one particular direction and not uniform in all directions.

3) Lambert's Law forms the basis of the Lambertion reflection; the law says that intensity of light is directly proportional to the angle between incident ray and the normal.

Check Your Progress 4

1) Lesser the value of Specular reflection parameter conveys the message that the surface under exposure is quite dull, the value of n increases as the surface shines more and more.

2) Gourand Shading.

3) Refer to Shading (8.4).

4) Say < refers to increasing order of comparative performance then constant shading < Gourand Shading < Phong Shading for reason behind refer to merits and demerits of each type of shading.

5) Refer to shading section.

6) Phong shading is better than Gourand shading, the reason being that in Gouraud Shading $\because$ of the consideration of average normal, the intensity is uniform across the edge between two vertices and this causes problem in shading. Deficiencies of Gourand Shading are overcome in Phong Shading, where we are finding intensities at different points across the edge and it is observed that intensity is varying non- uniformly across the edge between two vertex points, but this variation is uniform in Gouraud Shading. Thus Phong shading gives much better effect.

7) Resulting intensity (I) = $I_a K_a + I_d K_d \cos\theta + I_s K_s \cos^n \alpha$;

8) When an object is under the exposure of light, then the rays of light are distributed over the surface and the distribution of intensity pattern depends very much on the shape of object. Now to represent such 3D scene on computer we need to do rendering, i.e., transforming the 3D image into a 2D image, and because of this rendering there is a possibility of loss of information like depth, height etc., of an object in the scene. So to preserve this takes different Illumination models into consideration, for preserving the information embedded in 3D scene and let it not be lost while transforming it into 2D scene.

Check Your Progress 5

1) Ray tracing is a method of generating realistic images by computer, in which the paths of individual rays of light are followed from the viewer to their points of origin". Whereas ray casting only concerns itself with finding the visible surfaces of objects, ray tracing takes that a few steps further and actually tries to determine what each visible surface looks like.

2) Ray tracing makes use of the actual physics and mathematics behind light. Thus, the images produced can be strikingly life-like, or "photo-realistic". For more details refer to ray tracing algorithm.

3) If you put a grid over the image and only color those squares that are entirely within the circle, you will get something like next figure. This blockiness is called "aliasing", and is exactly what happens when you try to display a circle on a computer screen. The problem may be simplified by using a finer grid. Anti-aliasing is a method for improving the realism of an image by removing the jagged edges from it.