Chapter 5: Visible Realism

Credit hours:

Contents

5.1 Hidden Surface Removal

5.1 Visible Surface Detection

- Visible surface detection or Hidden surface removal
- When we view a picture containing non-transparent objects and surfaces, then we cannot see those objects from view which are behind from objects closer to eye.
- We must remove these hidden surfaces to get a realistic screen image.
- The identification and removal of these surfaces is called Hidden-surface problem.
- When we want to display a 3D object on a 2D screen, we need to identify those parts of a screen that are visible from a chosen viewing position.
- **Hidden surface removal or visible surface determination is the process used to determine which surfaces and parts of surfaces are not visible from a certain viewpoint.**
- Hidden surface determination is necessary to render an image correctly, so that one may not view features hidden behind the model itself, allowing only the naturally viewable portion of the graphics to be visible.

- It is major concern for realistic graphics for identifying those parts of a scene that are visible from a chosen viewing position.
- Visible surface detection methods are broadly classified according to whether they deal with objects or with their projected images.
  - Object-Space Methods (OSM)
  - Image-Space Methods (ISM)

Approaches for VSD

1. **Object-Space Methods (OSM):**
   - It compares objects and parts of objects to each other **within scene definition** to determine which surfaces are visible.
   - Deals with object definition.
   - E.g. Back-face detection method
2. **Image-Space Methods (ISM):**
   - Visibility is decided point by point at each pixel position **on the projection plane.**
   - Deals with projected image.
   - Most visible-surface algorithms use image-space methods.
   - E.g. Depth-buffer method, Scan-line method, Area-subdivision method

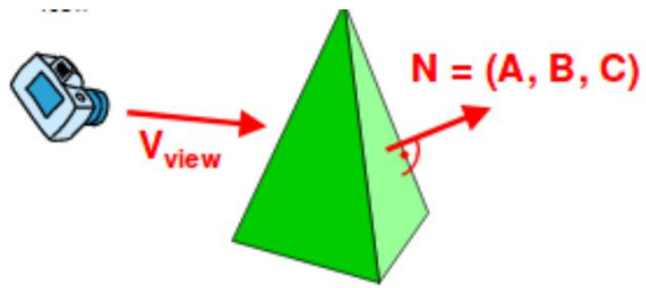
Assignment

- Object Space Method Vs Image space Method

Reference: https://www.javatpoint.com/computer-graphics-hidden-surface-removal

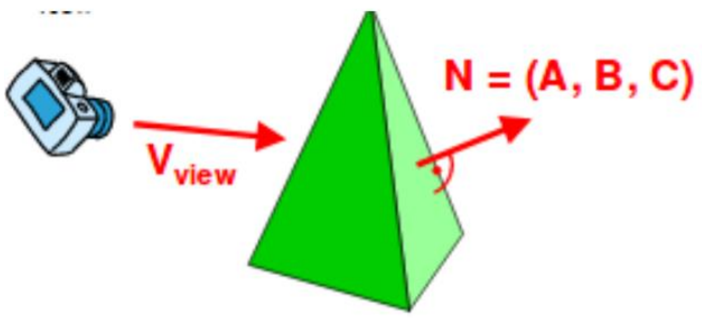
1 Back-face detection (Plane equation method)

- In a solid object, there are surfaces which are facing the viewer (front faces) and there are surfaces which are opposite to the viewer (back faces).
- Since we cannot see these surfaces anyway, to save processing time, we can remove them before the clipping process with a simple test.
- **Back-Face detection**, also known as **Plane Equation method,** is an **object space method** in which objects and parts of objects are compared to find out the visible surfaces.



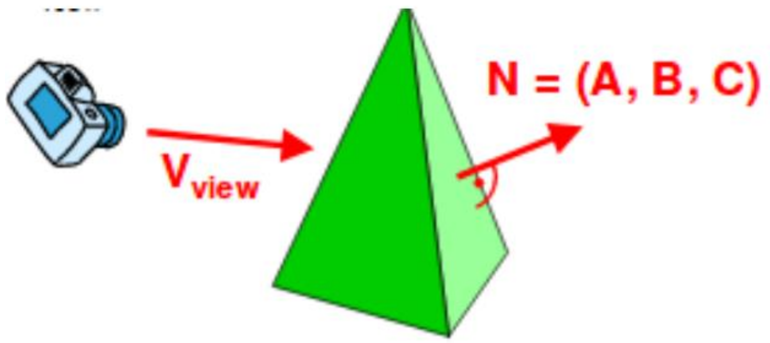
1 Back-face detection (Plane equation method)

- Let us consider a triangular surface that whose visibility needs to decided. Each surface has a normal vector.
  - If this **normal vector is pointing in the direction of the center of projection**, then it is a **front face** and can be seen by the viewer.
  - If this **normal vector is pointing away from the center of projection**, then it is a **back face** and can not be seen by the viewer.
- The test is very simple,
  - If the z component of the normal vector is positive,
    then, it is a back face.
  - If the z component of the vector is negative,
    it is a front face.



Principle:

- Remove all surfaces pointing away from the viewer.
- Eliminate the surface if it is completely obscured by other surfaces in front of it
- Render only the visible surfaces facing the viewer
- Back facing and front facing faces can be identified using the sign of V •N where V is the view vector and N is normal vector.
  - ➢ If V • N > 0, back face
  - ➢ if V • N < 0 front face
  - ➢ if V • N= 0 on line of view



Algorithm

Repeat for all polygons in the scene.

1. Do numbering of all polygons in clockwise direction i.e.

   $v_1\ v_2\ v_3 \ldots\ldots v_z$

2. Calculate normal vector i.e. N_1

   $N_1=(v_2-v_1)*(v_3-v_2)$

3. Consider projector P, it is projection from any vertex, Calculate dot product

   Dot=N.P

4. Test and plot whether the surface is visible or not.

   If Dot <0 then

   surface is visible

   else

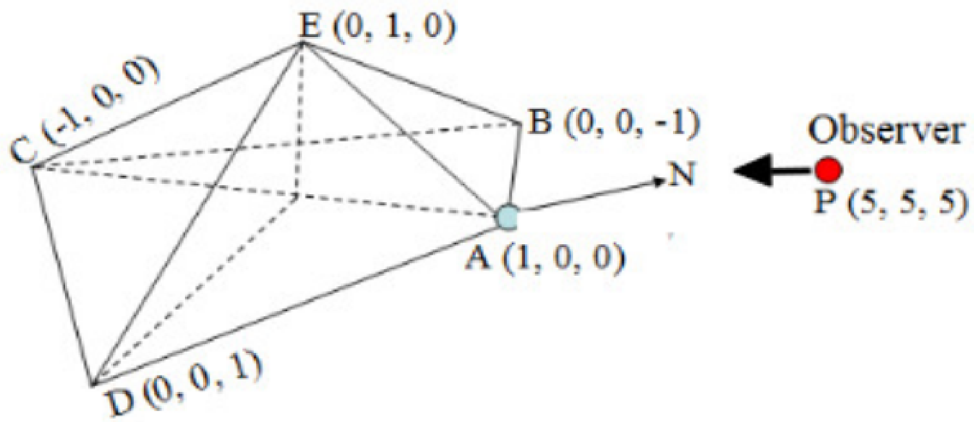
   Not visible

Numerical:
1. Find the visibility for the surface AED where an observer is at P (5, 5, 5)

Here,

$AE = (0-1)\,i + (1-0)\,j + (0-0)\,k = -i + j$

$AD = (0-1)\,i + (0-0)\,j + (1-0)\,k = -i + k$



Step-1: Normal vector N for AED

$$\text{Thus, } N = AE \times AD = \begin{pmatrix} i & j & k \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} = i\,(1-0) - j\,(-1+0) + k\,(0+1) = i + j + k$$

Step-2: If observer at P(5, 5, 5) so we can construct the view vector V from view point A(1,0, 0) as:

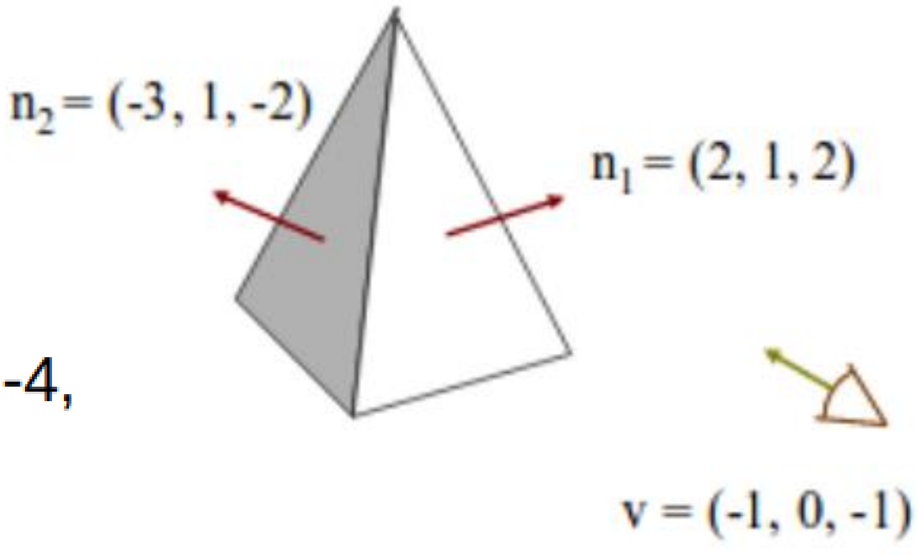
$V = PA = (1-5)\,i + (0-5)\,j + (0-5)\,k = -4i - 5j - 5k$

Step-3: To find the visibility of the object, we use dot product of view vector V and normal vector N as:

$V \bullet N = (-4i - 5j - 5k) \bullet (i + j + k) = -4-5-5 = -14 < 0$

This shows that the surface is visible for the observer.

2. Find the visibility of n1 and n2 from V.



n1 • v = (2, 1, 2) ·(-1, 0, -1) = -2 – 2 = -4,

i.e n1 • v< 0

So, n1 front facing polygon

n2 • v = (-3, 1, -2) ·(-1, 0, -1) = 3 + 2 = 5

i.e n2 • v > 0

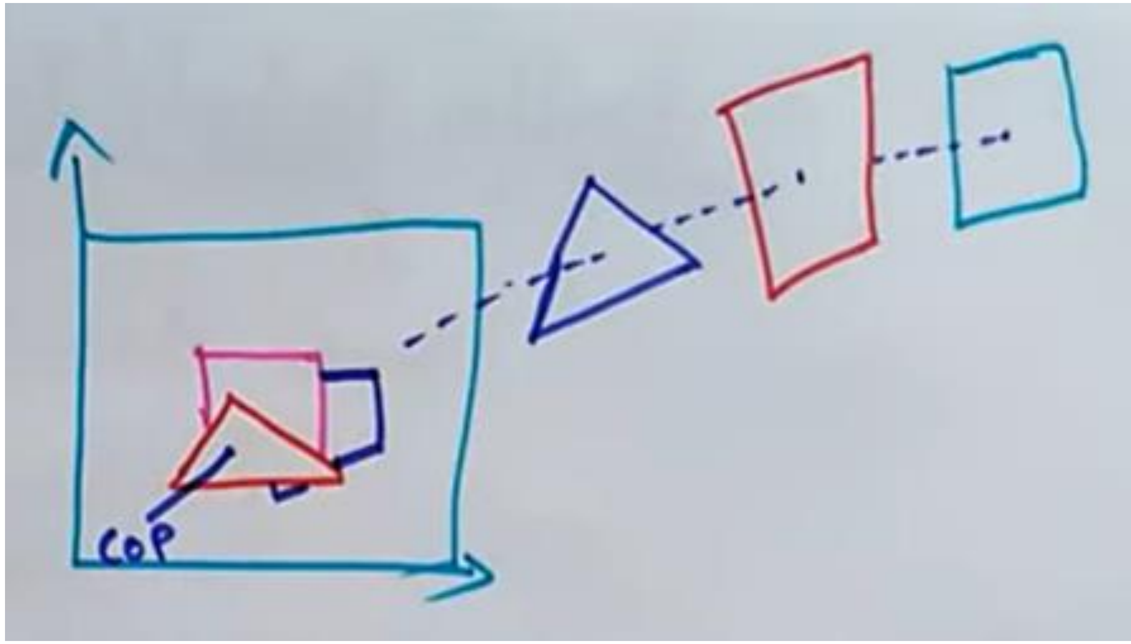
so n2 back facing polygon

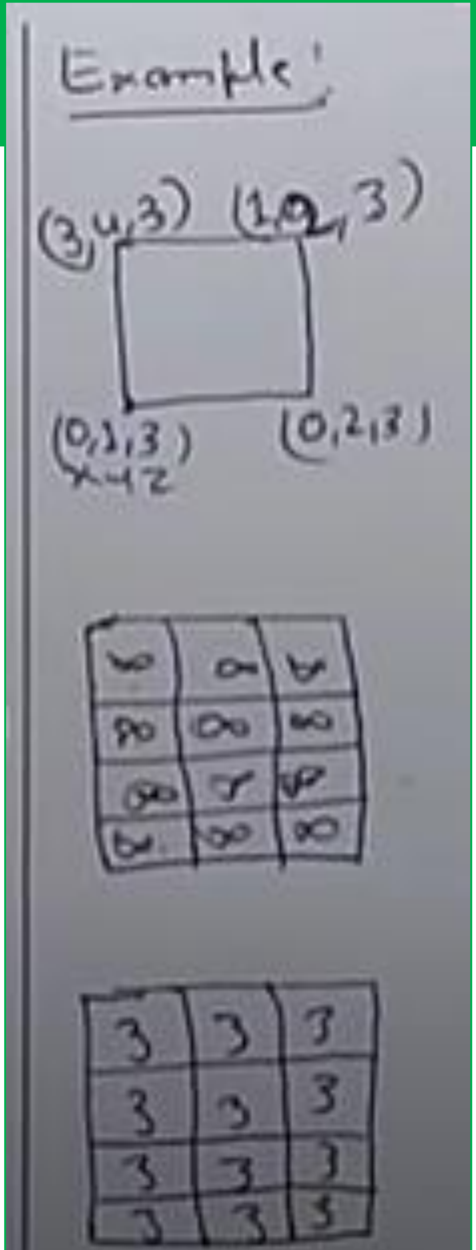
- **Advantages:**
- It is a simple and straight forward method.
- It reduces the size of databases, because no need of store all surfaces in the database, only the visible surface is stored.

- **Limitations:**
- This method works fine for convex polyhedral, but not necessarily for concave polyhedral or overlapping objects. So, we need to apply other methods to further determine where the obscured faces are partially or completely hidden by other objects (e.g. Using Depth-Buffer Method or Depth-sort Method).
- This method can only be used on solid objects modeled as a polygon mesh.

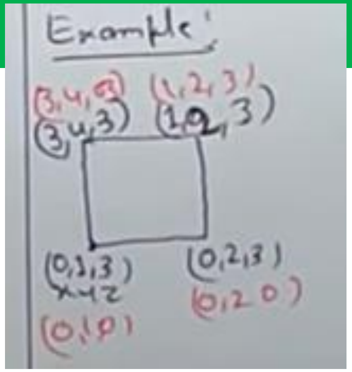
2. Depth-buffer method (Z-Buffer Method)

- Image space approach
- Implemented in screen coordinate system.
- Basic idea is to test z-depth of each surface to determine the closest (visible) surface.

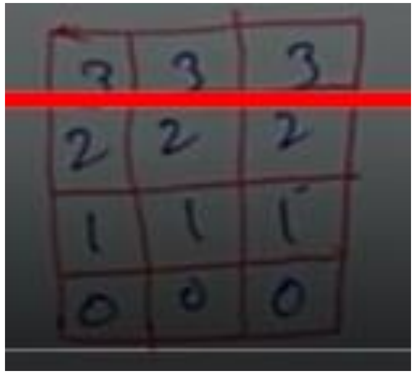




Example:
(3,4,3) (1,2,3)
(0,1,3) (0,2,3)
xyz
3 3 3
3 3 3
3 3 3
3 3 3

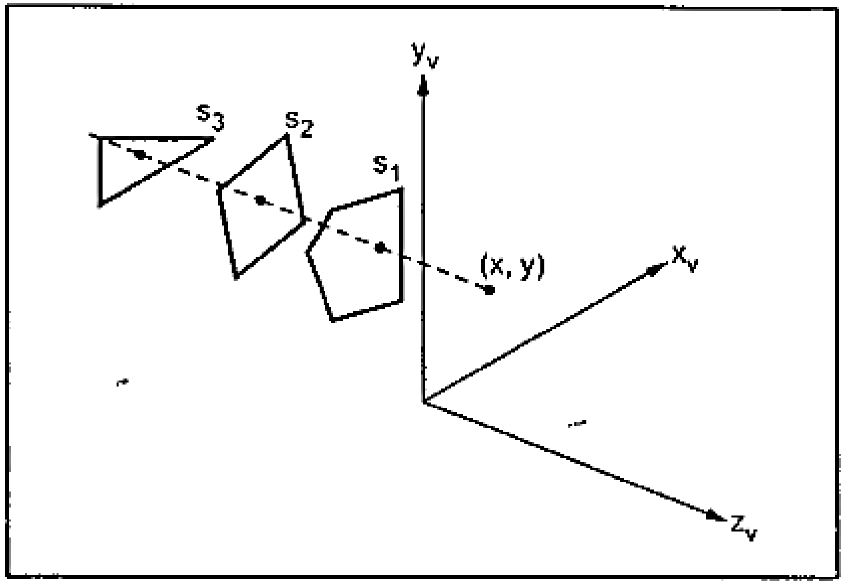


Example:
(3,4,0) (1,2,3)
(3,4,3) (1,2,3)
(0,1,3) (0,2,3)
xyz
(0,2,0)
(0,1,0)



3 3 3
2 2 2
1 1 1
0 0 0

- It is **an image space method** for detecting visible surface.
- The Z-buffer method is a method that **compares surface depths of each pixel** position on the projection plane. Normally, z-axis is represented as depth.
- Each surface of a scene is processed separately, one point at a time across the surface. And each (x, y, z) position on a polygon surface corresponds to the projection point (x, y) on the view plane.
- In this method, two buffers are used:
  1. **Depth Buffer (Z-buffer):** Stores depth values for each (x, y) position.
  2. **Frame Buffer (refresh buffer):** Stores the intensity values for each position.
- For example, in figure shown below, among three surfaces, surface S1 has the smallest depth at view position (x, y) and hence highest z value. So it is visible at that position.



- **Advantages:**
  - Easy to implement
  - Reduces the speed problem if implemented in hardware
  - Processes one object at time.
  - Accurate performance.
- **Disadvantages:**
  - Requires large memory.
  - Time consuming process.

Algorithm:

- **Step-1 – Set the buffer values**
  - Depthbuffer (x, y) = 0
  - Framebuffer (x, y) = background color
- **Step-2 – Process each polygon surface (One at a time)**
  - For each projected (x, y) pixel position of a polygon, calculate depth z.
    i.e. $Z = \frac{-Ax-By-D}{C}$
  - If Z > depthbuffer (x, y)
    - Compute surface color,
    - set depthbuffer (x, y) = z,
    - framebuffer (x, y) = surfacecolor (x, y)
- **Step-3** - After all pixels and surfaces are compared, **draw object using X,Y,Z** from depth and intensity refresh buffer.

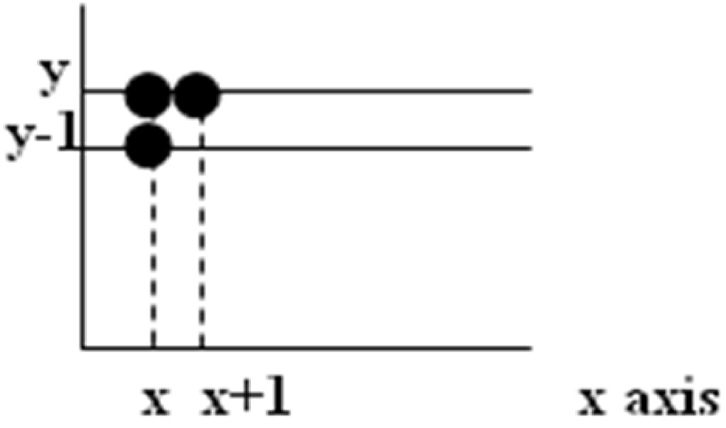
- Initially, all positions in the **depth buffer are set to 0** (minimum depth), and **the refresh buffer is initialized to the background intensity.**
- Each surface listed in the polygon tables is then processed, one scan line at a time, **calculating the depth (z-value) at each (x, y) pixel** position.
  - ➢ To calculate z-values, the plane equation $A x + B y + C z + D = 0$ is used where (x, y, z) is any point on the Plane, and the coefficient A, B, C and D are constants describing the spatial properties of the Plane.
  - ➢ Therefore, we can write: $Z = \frac{-Ax-By-D}{C}$
- The **calculated depth is compared to the value previously stored** in the depth buffer at that position.
  - ➢ If the calculated depth is **greater than the value stored** in the depth buffer,
    - o **the new depth value is stored**, and
    - o the **surface intensity at that position is determined** and
    - o placed in the same *xy location in the refresh buffer.*
- This processing is done one scan line at a time.
- Then for any scanline, the adjacent horizontal (x) and vertical (y) value on the scanline both differs by 1.

- Then, at successive pixel (x + Δx, y) and since Δx=1, the value of Z' at next position (x+1, y) along the scan line is

  $$Z' = \frac{-A(x+1) - By - D}{C}$$

  ie $Z' = Z - \frac{A}{C}$

  where, $-\frac{A}{C}$ is the constant for each surface.



- A similar incremental calculation can be performed to determine the first value of Z on the next scan line, decrementing by B/C for each Δy.

  $$Z = \frac{-A - B(y-1) - D}{C}$$

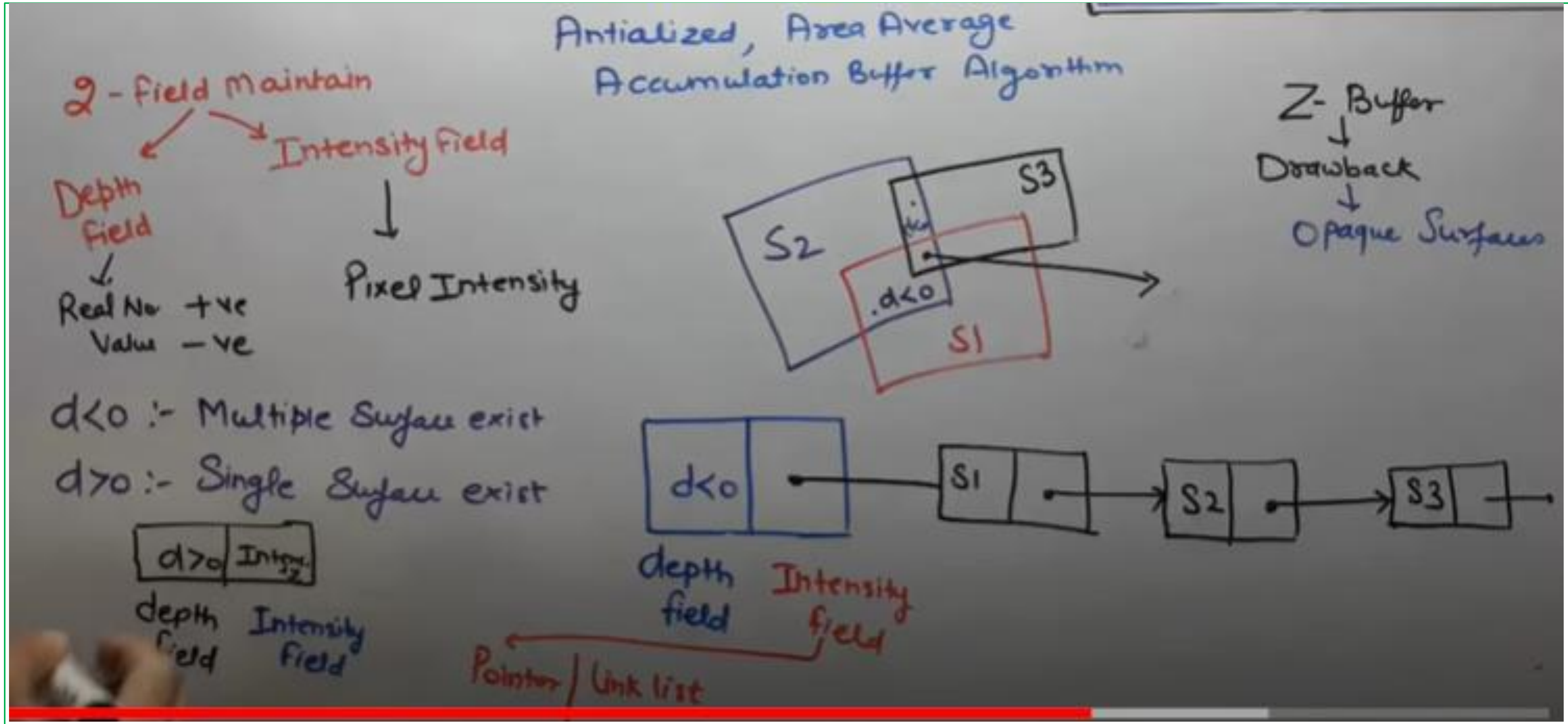
- Then, at successive pixel (x + Δx, y- Δy) and since Δx=1 and Δy=-1 the value of Z' at next position (x+1, y-1) along the next scan line is

  $$Z' = \frac{-A(x+1) - B(y-1) - D}{C}$$

  ie $Z' = Z - \frac{A}{C}$

  where, $-\frac{A}{C}$ is the constant for each surface in the next scan line.

3. A-Buffer Method

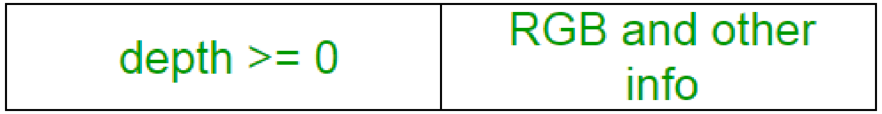


Reference: https://www.youtube.com/watch?v=m5vT2dlEfO4

3. A-Buffer Method

- A-Buffer method in computer graphics is a general hidden face detection mechanism suited to medium scale virtual memory computers.
- This method is also known as **anti-aliased** or **area-averaged** or **accumulation buffer**.
- This method extends the algorithm of depth-buffer (or Z Buffer) method.
- As the depth buffer method can only be used for opaque object but not for **transparent object**, the A-buffer method provides advantage in this scenario.
- Although the A buffer method requires more memory, but different surface colors can be correctly composed using it.
- Being a descendant of the Z-buffer algorithm, each position in the buffer can reference a linked list of surfaces.
- The **key data structure** in the A buffer is the **accumulation buffer**.

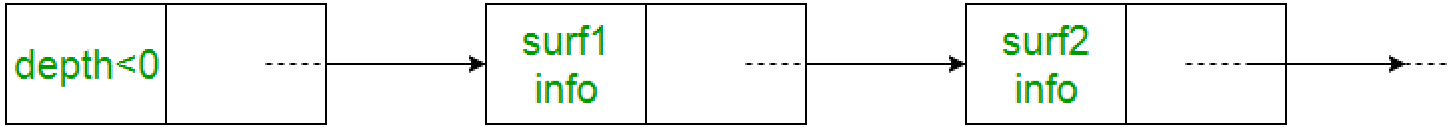
- Each position in the A buffer has 2 fields :
  1) Depth field: It stores a positive or negative real number.
  2) Surface data field or Intensity field: It stores surface intensity information or a pointer value to a linked list of surfaces that contribute to that pixel position.



(a) When a pixel overlap by only one surface

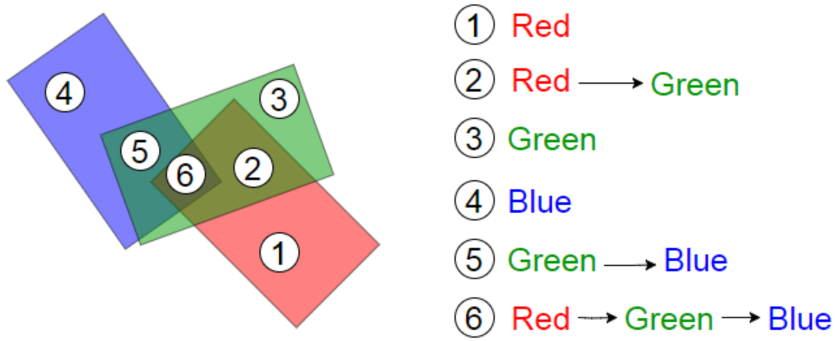
- If the value of depth is >= 0, the number stored at that position is the **depth of single surface** overlapping the corresponding pixel area.

- If depth<0, it indicates multiple-surface contributions to the pixel intensity. The intensity field then stor



(b) When a pixel overlaps by multiple surfaces

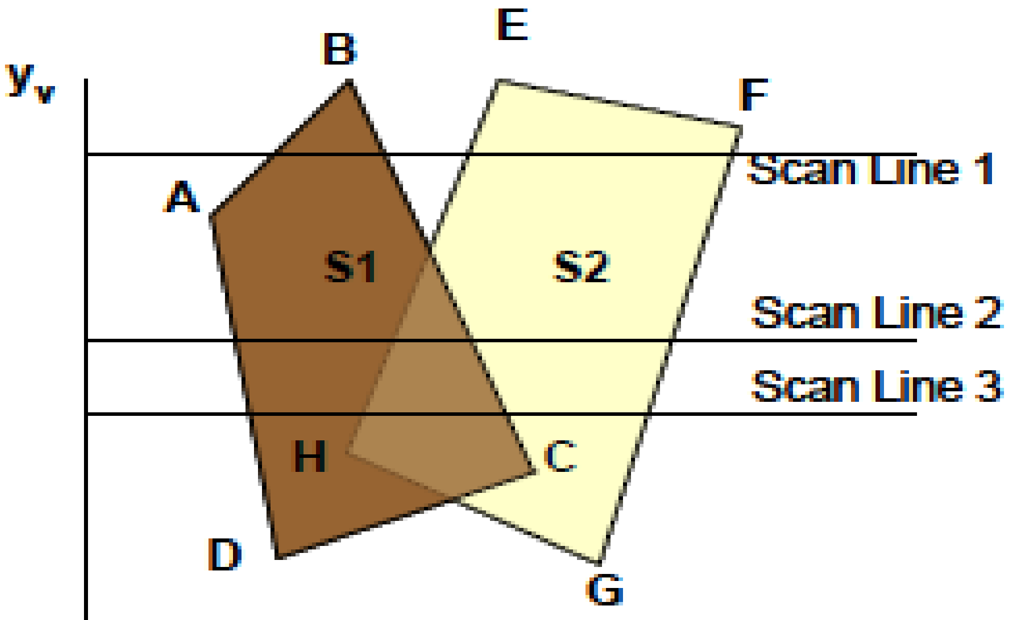
- A buffer method is slightly costly than Z-buffer method because it requires more memory in comparison to the Z-buffer method.
- It proceeds just like the depth buffer algorithm. Here, the depth and opacity are used to determine the final color of the pixel.
- The surface buffer in the A buffer method includes :
  1. Depth
  2. Surface Identifier
  3. Opacity Parameter
  4. Percent of area coverage
  5. RGB intensity components
  6. Pointer to the next surface



- In A-buffer method, each pixel is made up of a group of sub-pixels. The final color of a pixel is computed by summing up all of its sub-pixels.

4. Scan Line Method

- It is an image-space method to identify visible surface.
- This method is an **extension of scan line algorithm** for filling polygon interiors. Instead of filling just one surface, we deal with multiple surfaces.
- As each scan line is processed, all polygon surfaces intersecting that line are examined to determine which are visible.
- Two important tables, edge table and polygon table, are maintained for this.
  - **The Edge Table** − It contains coordinate endpoints of each line in the scene, the inverse slope of each line, and pointers into the polygon table to connect edges to surfaces.
  - **The Polygon Table** − It contains the plane coefficients, surface material properties, other surface data, and may be pointers to the edge table.



For scan line 1

- The active edge list contains edges AB,BC,EH, FG
- Between edges AB and BC, only *flags for s1 == on and between edges EH and FG, only flags for s2==on*
- No depth calculation needed and corresponding surface intensities from S1 and S2 both are entered in refresh buffer.

Note :

- The edges are sorted in order of increasing x.
- Define flags for each surface to indicate whether a position is inside or outside the surface

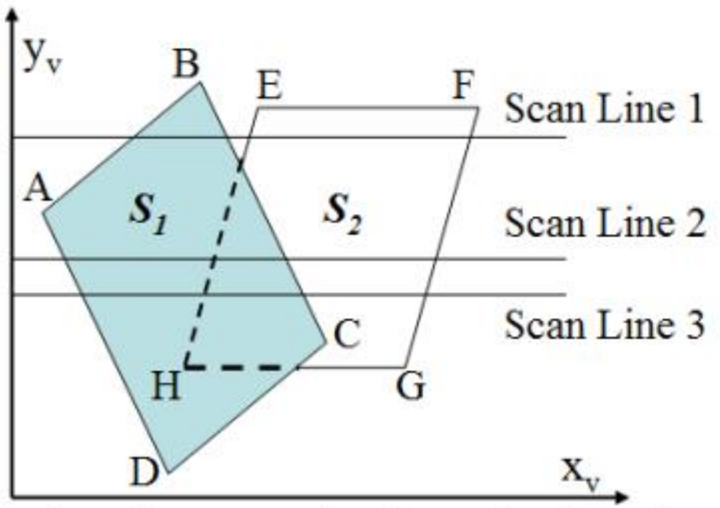


Fig. Scan lines crossing the projection of two surfaces, S_1 and S_2 in the view plane. Dashed lines indicate the boundaries of hidden surfaces.

For scan line 2

- The active edge list contains edges AD,EH,BC and FG
- Between edges AD and EH, only the *flag for surface s1 == on*
- Between edges EH and BC *flags for both surfaces == on*
- Depth calculation (using plane coefficients) is needed using the plane coefficients for the two surfaces.
- For this example, the depth of surface S1 is assumed to be less than that of S2 , so intensities for surface S1 are loaded into the refresh buffer until boundary BC is encountered.
- Then the flag for surface S1 goes off, and intensities for surface S2 are stored until edge FG is passed.

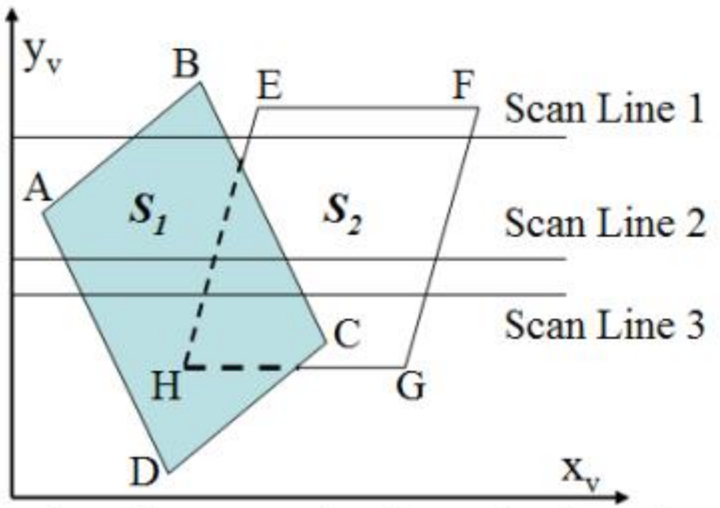
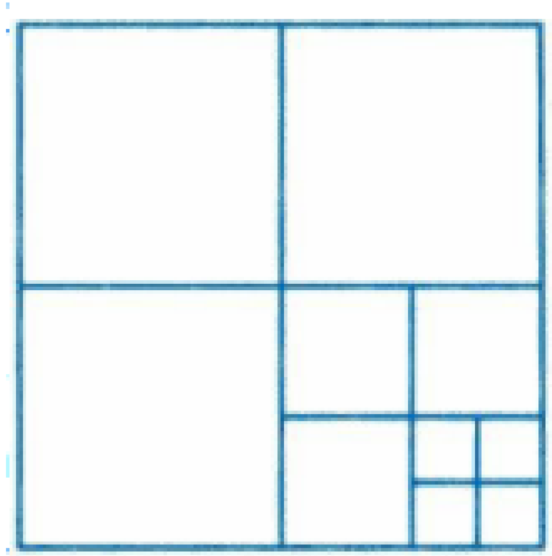


Fig. Scan lines crossing the projection of two surfaces, S_1 and S_2 in the view plane. Dashed lines indicate the boundaries of hidden surfaces.

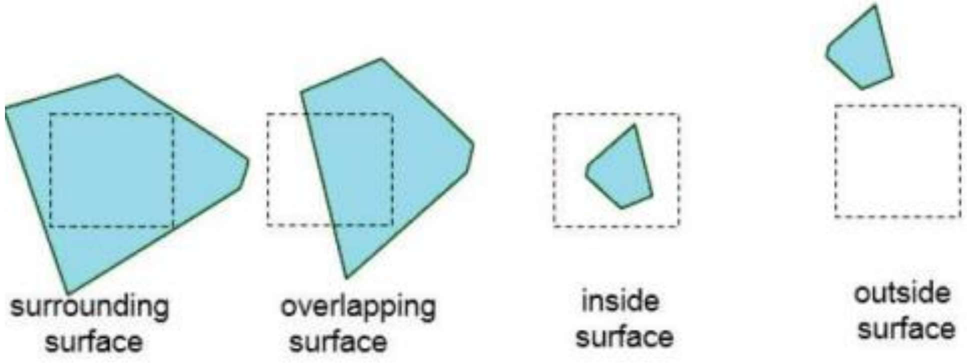
5. Area-Subdivision Method

- The area-subdivision method takes advantage by locating those view areas that represent part of a single surface.
- Divide the total viewing area into smaller and smaller rectangles until each small area is the projection of part of a single visible surface or no surface at all.
- Continue this process until the subdivisions are easily analyzed as belonging to a single surface or until they are reduced to the size of a single pixel.
- An easy way to do this is to successively divide the area into four equal parts at each step.



- There are four possible relationships that a surface can have with a specified area boundary.

1. **Surrounding surface** − One that completely encloses the area.
2. **Overlapping surface** − One that is partly inside and partly outside the area.
3. **Inside surface** − One that is completely inside the area.
4. **Outside surface** − One that is completely outside the area.



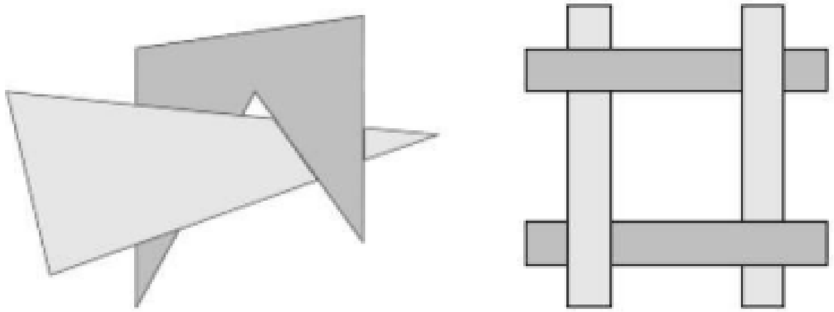
- The tests for determining surface visibility within an area can be stated in terms of these four classifications. No further subdivisions of a specified area are needed if one of the following conditions is true −
  - All surfaces are outside surfaces with respect to the area.
  - Only one inside, overlapping or surrounding surface is in the area.
  - A surrounding surface obscures all other surfaces within the area boundaries.

5. Depth Sorting Method (Painter’s A;gorithm)

- Depth sorting method uses both image space and object-space operations.
- This algorithm is also called "Painter's Algorithm" as it simulates how a painter typically produces his painting by starting with the background and then progressively adding new (nearer) objects to the canvas. • Thus, each layer of paint covers up the previous layer.
- The depth-sorting method performs two basic functions –
  - First, the surfaces are **sorted in order of decreasing depth.**
  - Second, the **surfaces are scan-converted** in order, starting with the surface of greatest depth. The scan conversion of the polygon surfaces is performed in image space.
- The **algorithm** begins by sorting by depth.
  - Sort all surfaces according to their distances from the view point.
  - Render the surfaces to the image buffer one at a time starting from the farthest surface.
  - Surfaces close to the view point will replace those which are far away.
  - After all surfaces have been processed, the image buffer stores the final image

Problem:

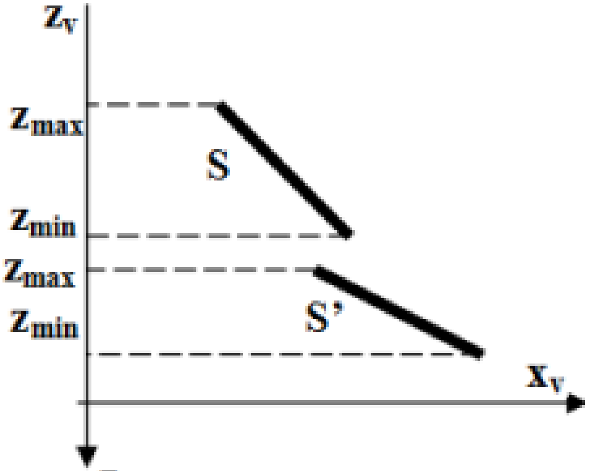
- One of the major problem in this algorithm is intersecting polygon surfaces as shown in figures below:



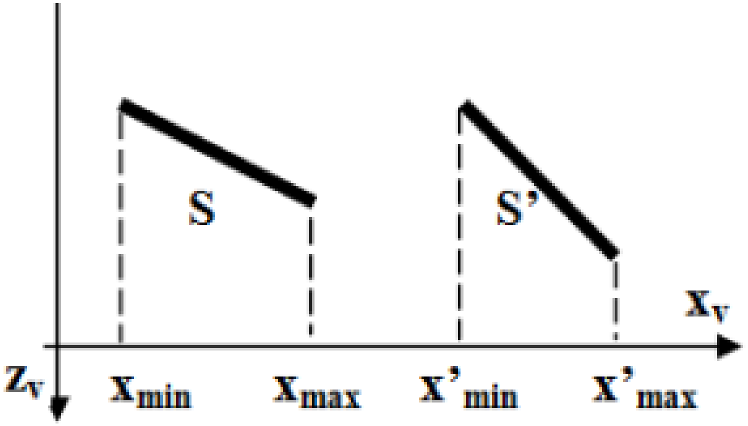
  - Different polygons may have same depth.
  - The nearest polygon could also be the farthest one.
- So, we cannot simply use depth-sorting to remove the hidden surfaces in the images.

Example:

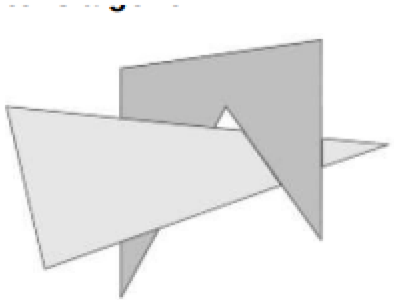
- Assuming we are viewing along the z axis.
- Surface S with the greatest depth is then compared to other surfaces in the list to determine whether there are any overlaps in depth.
  - If no depth overlaps occur, S can be scan converted. This process is repeated for the next surface in the list.
- However, if depth overlap is detected, we need to make some additional comparisons to determine whether any of the surfaces should be reordered.

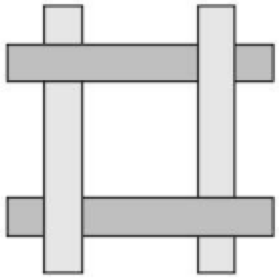


No Depth Overlap



Depth overlap but no overlap in x direction





References:

- https://studynotesnepal.com/wp-content/uploads/2023/02/CG-7.pdf
- https://gsprasadareddy.files.wordpress.com/2019/10/unit_5_visible_surface_detection_methods.pdf
- https://www.cvs.edu.in/upload/Lec%202.30-3.30%2006.05.2020.pdf
- https://www.slideshare.net/anku2266/visible-surface-detection-35177585

5.2 Illuminations and Shading

5.2 Illumination theory

Illumination and its need:

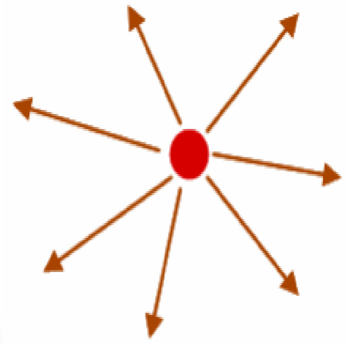
- Illumination is an observable property and effect of light. The placement of light sources can make a considerable difference in the type of message that is being presented.
- Multiple light sources can wash out any wrinkles in a person's face, for instance, and give a more youthful appearance.
- In contrast, a single light source, such as harsh daylight , can serve to highlight any texture or interesting features.
- For realistic displaying of 3D scene it is necessary to calculate appropriate color or intensity for that scene.
- Components of Illumination model
    - Light Sources: type, color, and direction of the light source
    - Surface Properties : reflectance, opaque/transparent, shiny/dull.

Factors in Illumination Model

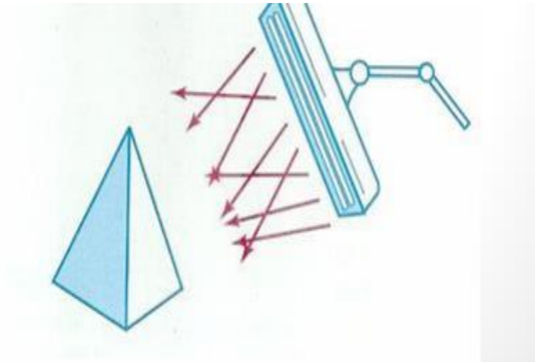
- Illumination models involve number of factors like:
  1. Optical properties of the surfaces (transparency, reflectivity, surface texture)
  2. Relative positions of the surfaces in a scene.
  3. Color and position of the light sources, and
  4. Position and orientation of the viewing plane.

Light sources

- **Light Source :**
  Light source is the light emitting source. There are three types of light sources. Their position, electromagnetic spectrum and shape determine the lightning effect.
  a) **Point Sources –** All light rays originate at a point and radially diverge. The source that emit rays in all directions (A bulb in a room).
  b) **Parallel Sources –** Light rays are parallel. Can be considered as a point source which is far from the surface (The sun).
  c) **Distributed Sources –** All light rays originate from a finite area. It models a nearby source. (A tube light).



Point Light Source



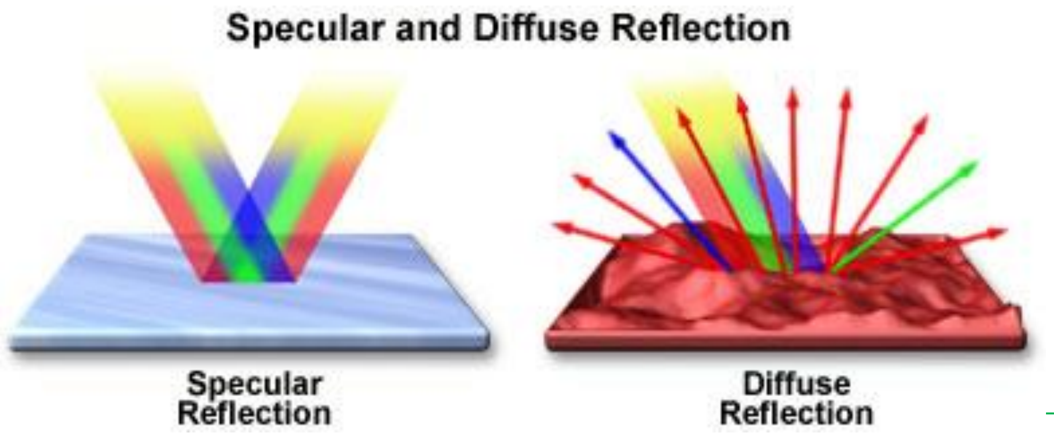
Distributed Light Source

Surface Rendering

- **Surface rendering** means a procedure for applying a lighting model to obtain pixel intensities for all the projected surface positions in a scene.
- A surface rendering algorithm uses the intensity calculations from an illumination model to determine the light intensity for all projected pixel positions for the various surfaces in a scene.
- Surface rendering can be performed by applying the illumination model to every visible surface point.

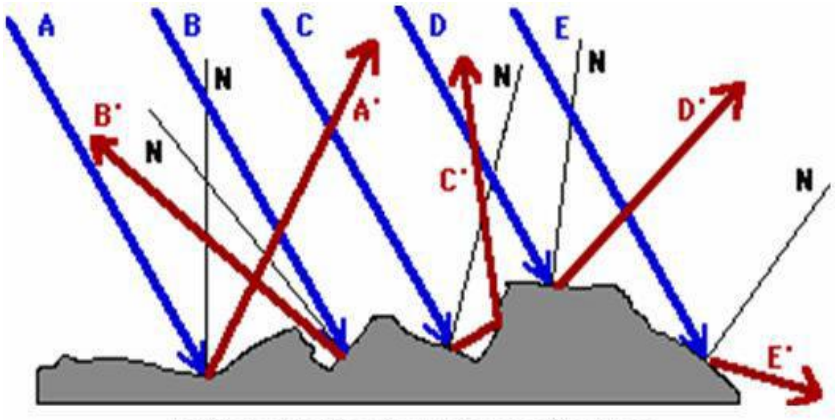
5.2.1 Reflection of light:

- When light is incident on opaque surface, part of it is reflected and part of it is absorbed. ∴ $I = A+R$
- The amount of incident light reflected by a surface depends on the type of material.
  - **Shining material reflects** more incident light and **dull surface absorbs** more of the incident light.
  - For transparent surfaces, some of the incident light will be reflected and some will be transmitted through the material.



i) Diffuse Reflection

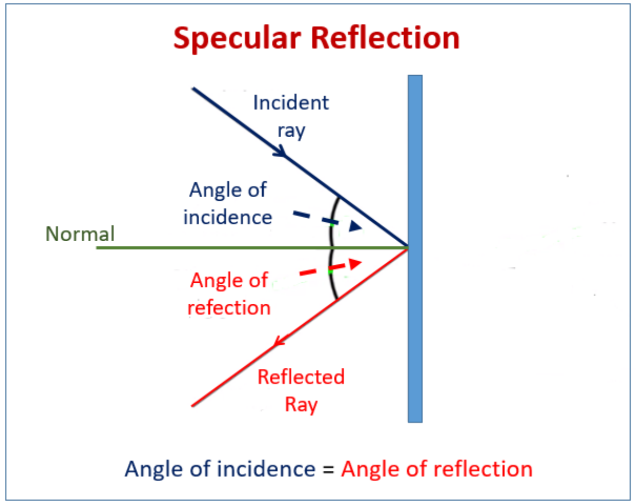
- Diffuse reflection is the reflection of light from a surface such that a ray incident on the surface is scattered at many angles rather than at just one angle as in the case of specular reflection.
- An ideal diffuse reflecting surface is said to exhibit Lambertian reflection, meaning that there is equal luminance when viewed from all directions lying in the half-space adjacent to the surface.



Diffuse Reflection From a Surface

ii) Specular reflection

- Specular or regular reflection is the mirror-like reflection of waves, such as light, from a surface.
- In this process, each incident ray is reflected at the same angle to the surface normal as the incident ray, but on the opposing side of the surface normal in the plane formed by incident and reflected rays.
- The result is that an image reflected by the surface is reproduced in mirror-like (specular) fashion.

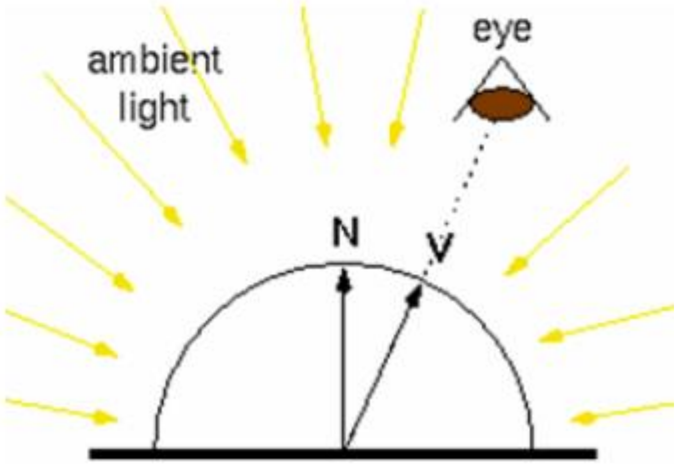


5.2.3 Illumination Models

- Illumination model is method for calculating light intensities.
- Light calculations are based on the optical properties of the surface, the background lightning conditions and the light-source specifications.
- Optical parameters are used to set surface properties such as transparency, opacity, etc. and these control the amount of reflection and absorption of incident light.
- All light sources are considered to be point source specified with a coordinate position and intensity value (color).
- Three basic models:
  1. Ambient Light
  2. Diffuse Reflection
  3. Specular Reflection

i) Ambient lighting Model

- A surface that is not exposed directly to a light source still will be visible if nearby objects are illuminated.
- In our basic illumination model, we can set a general level of brightness for a scene.
- This is a simple way to model the combination of light reflections from various surfaces to produce a uniform illumination called the ambient light, or background light.
- Ambient light has no spatial or directional Characteristics. The amount of ambient light incident on each object is a constant for all surfaces and over all directions.
- Ambient (or Background) light means **the light that is already present in a scene**, before any additional lighting is added.
- It usually refers to **natural light**, either outdoors or coming through windows, etc.It can also mean **artificial lights** such as normal room lights. Not all ambient light is natural light, but all natural light is ambient light.



- The term "ambient light" is also used to describe the light already present in a room before any additional light is added.

- Ambient lighting is an important element of computer graphics because it helps to create a realistic environment in an image. Without ambient lighting, an image can appear flat and unappealing. Ambient lighting adds a sense of depth and realism to an image, making it appear more life-like. It also helps to create the illusion of a 3D space, which is essential for creating realistic-looking images.

- 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:

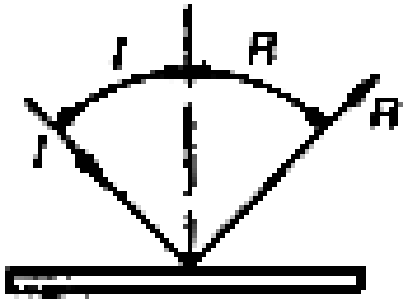
$$\mathbf{I=K_a}$$

Where $\mathbf{I}$ is the resulting intensity and $\mathbf{K_a}$ is the object's intrinsic intensity.

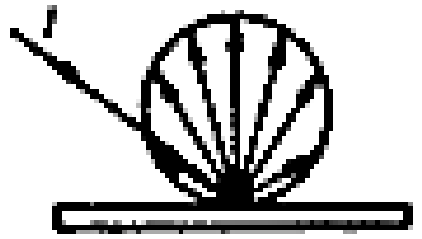
- If we assume that ambient light impinges equally on all surface from all direction, then $\mathbf{I = I_a\, K_a}$
  - Where $\mathbf{I_a}$ is intensity of ambient light. $\mathbf{K_a}$ is object's intrinsic intensity.
- The amount of light reflected from an object's surface is determined by $\mathbf{K_a}$, the ambient-reflection coefficient $\mathbf{K_a}$ ranges from 0 to 1.

ii) Diffuse Reflection Model

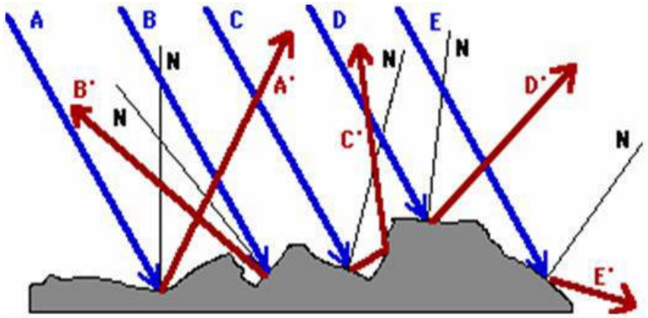
- It is the characteristics of light reflected from a dull, non-shiny surface.
- Diffuse reflections are constant over each surface in a scene in a scene, independent of the viewing direction. This means position of viewer is not important.
- Such surfaces are called ***ideal diffuse reflectors (also referred to as Lambertian reflectors)***
- When such illumination is uniform form all directions, it is called Diffuse reflection.
- Diffuse reflection is the reflection of light from a surface such that an incident ray is reflected at many angles, rather than at just one angle as in the case of specular reflect



DIRECT REFLECTION



DIFFUSED REFLECTION



Diffuse Reflection From a Surface

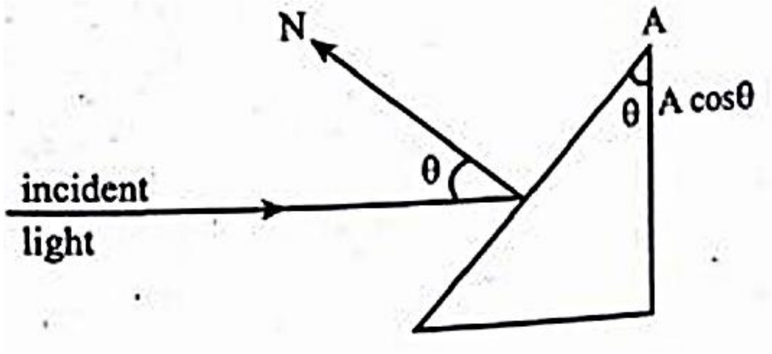
- Assuming diffuse reflections from the surface are scattered with equal intensity in all directions, applying Lemert's Cosine law, we have:

$$I_{diff} = K_d \, I_L \, \mathrm{Cos}\, \theta$$

  - Where I_L is the intensity of the point light source.
  - θ is angle of incidence between the incoming light direction and the surface normal

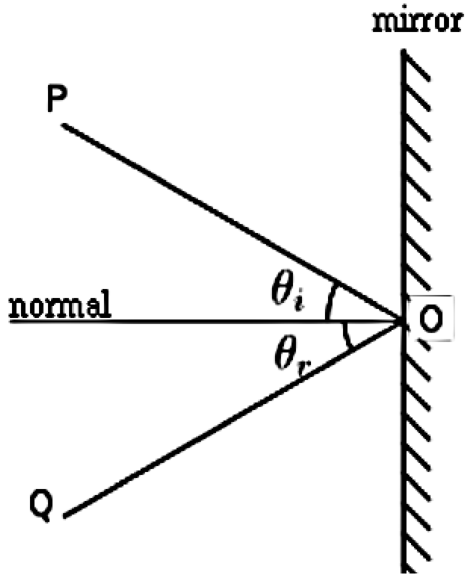
- If N is the unit normal vector to a surface and L is the unit vector in the direction to the point light source, then:

$$I_{L,\,diff} = K_d \, I_L \, (N.L)$$



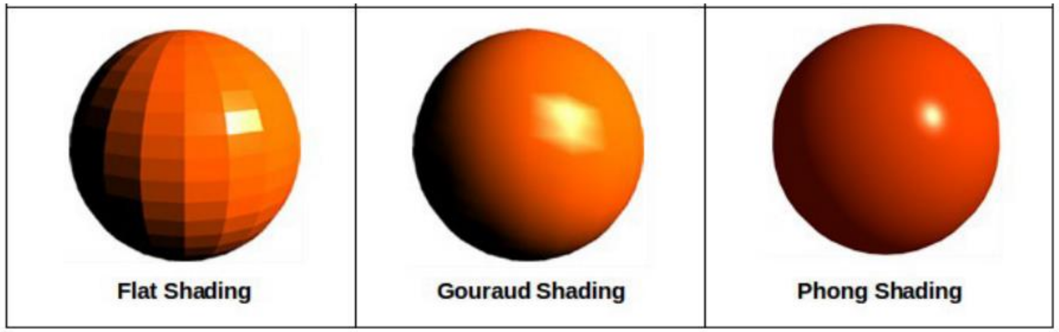
iii) Specular Reflection Model

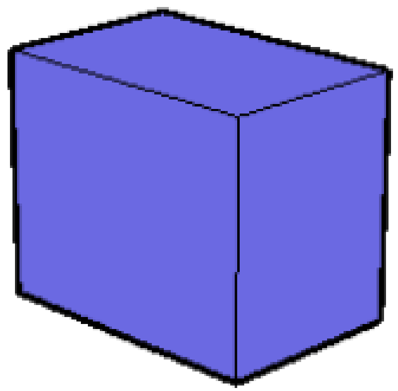
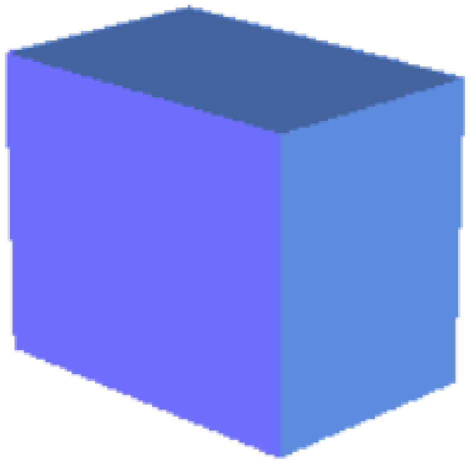
- Specular reflection is also known as mirror-like reflection or regular reflection
- Specular reflection is the mirror-like reflection of waves, such as light, from a surface.
- In this process, each incident ray is reflected at the same angle to the surface normal as the incident ray, but on the opposing side of the surface normal in the plane formed by incident and reflected rays.
- The result is that an image reflected by the surface is reproduced in mirror-like (specular) fashion.
- In specular reflection, the angle of the incident ray is equal to the angle of the reflected ray.
- If the reflecting surface is irregular it will create diffuse reflection, rather than specular reflection.



5.2.4 Surface Shading methods:

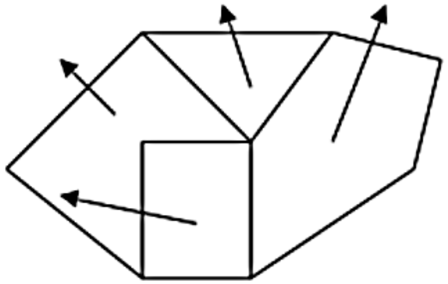
- In computer graphics, shading refers to the process of altering the color of an object/surface/polygon in 3D scene to create a photo with realistic effect.
- This is based on things like:
  - The surface's angle to lights,
  - its distance from lights,
  - its angle to the camera
  - Material properties (e.g. bidirectional reflectance distribution function)
- Shading is performed during the rendering process by a program called a shader.
- Surface Shading methods:
  a) Constant Shading
  b) Gouraud Shading
  c) Phong Shading
  d) Fast Phong Shading



		
Rendered image of a box. This image has no shading on its faces, but uses edge lines to separate the faces.	This is the same image with the edge lines removed.	This is the same image rendered with shading of the faces to alter the colors of the 3 faces based on their angle to the light sources.

a) Constant Shading (Flat Shading)

- It is fast and simple but cannot model specular reflection.
- A single intensity is calculate for each polygon and all points over the surface of the polygon are then displayed with the same intensity value.
- Useful for quickly displaying the general appearance of a curved surface.
- A sharp intensity discontinuation is seen in the border between two polygons.
- Provide an accurate rendering for an object if all of the following assumptions are valid:
  - ➢ The object is a polyhedron and is not an approximation of an object with a curved surface.
  - ➢ All light source illuminating the object are sufficiently far from the surface so that N.L and the attenuation function are constant over the surface.
  - ➢ The viewing position is constant over the surface.

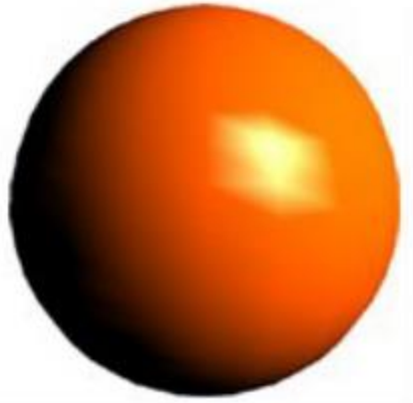


Flat Shading

b) Gouraud Shading

- Renders a polygon surface 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 discontinuation that can occur in flat shading.
- Produces more realistic results, but requires considerable more calculations.
- **Advantages:** Removes intensity discontinuities at the edge as compared to constant shading.
- **Disadvantage:** highlights on the surface are s.t. displayed with anomalous shap and linear intensity interpolation can cause bright or dark intensity streak calle mach-bands.
- **Steps:** It involves below calculation for each polygon surface:
  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.

Flat Shading



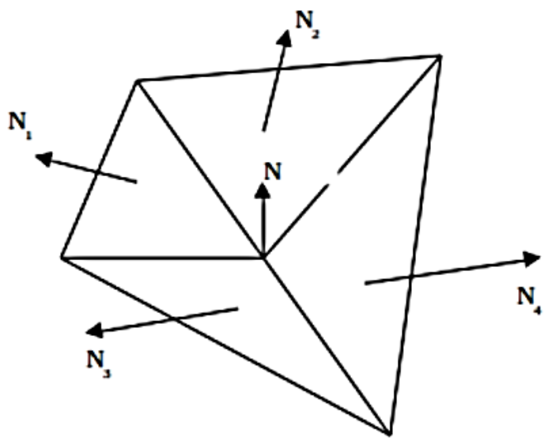
Gouraud Shading

For instance, calculation of an object with four surface, Calculation for each polygon surface:

i) Determining the average unit vector at each vertex:

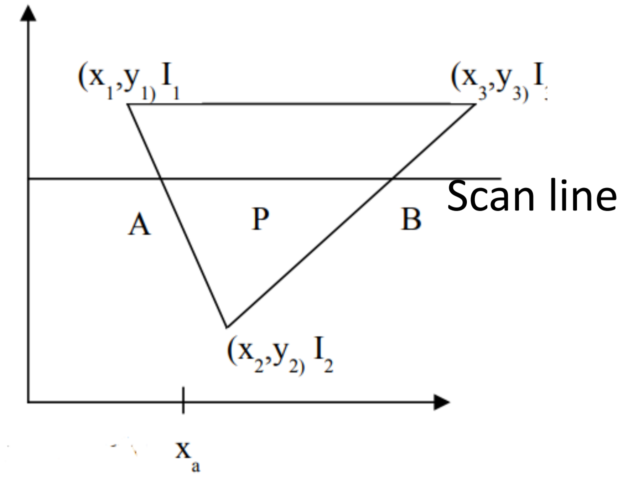
- At each polygon vertex, we obtain a normal vertex N_V by averaging the surface normal of all polygons sharing the vertex as:

$$N_v = \frac{N_1 + N_2 + N_3 + N_4}{|N_1 + N_2 + N_3 + N_4|}$$

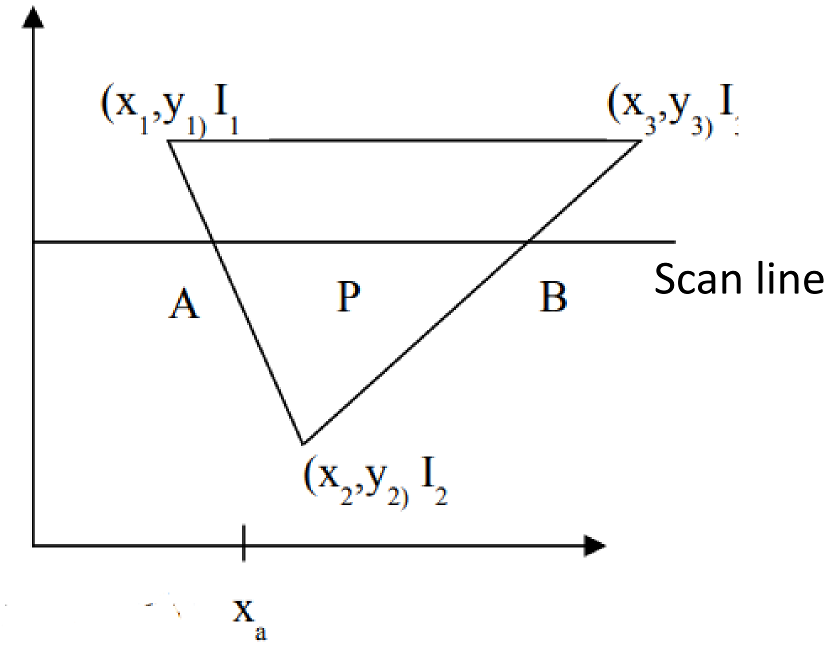


ii) Applying illumination model:

- Once N_v is known, intensity at the vertices can obtain from lighting model.
- Here in the figure, the intensities of vertices I1, I2, I3 are obtained by averaging normal of each surface sharing the vertices and applying a illumination model.



iii) Linear Interpolation:



- For each scan line, intensity at intersection of line with polygon edge are linearly interpolated from the intensities at the edge point.
- So, intensities at intersection point A, I_a is obtained by linearly interpolating intensities or I1 and I2 as:

$$I_a = \frac{y_a - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_a}{y_1 - y_2} I_2$$

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

$$I_b = \frac{y_a - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_a}{y_3 - y_2} I_2$$

- The intensities of a point P in the polygon surface along the scan line is obtained by linearly interpolating intensities I_a and I_b as:

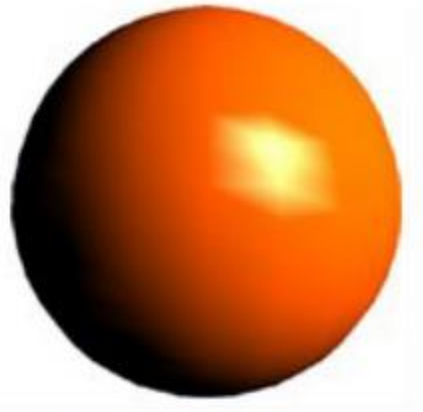
$$I_p = \frac{x_p - x_a}{x_b - x_a} I_b + \frac{x_b - x_p}{x_b - x_a} I_a$$

Mach band effect:

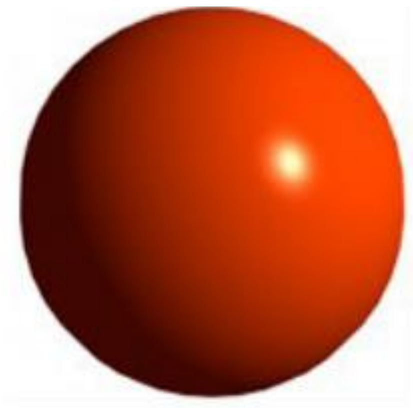
- Mach bands are an optical illusion where a band of gradients will appear in places to be lighter or darker than they actually are.
- When looking at Mach bands, one sees a central band of a light to dark gradient surrounded on one side by the lightest color and on the opposite side by the darkest color.
- Mach bands, as well as numerous other visual and perceptual illusions, help scientists study the way the eye and brain process visual information.

c) Phong Shading

- It is normal vector interpolation method.
- Greatly reduced mach-band effect, and produces more realistic surfaces.
- More accurate interpolation method which interpolate the vector normal over the surface of the polygon.
- **Advantage:** Displays more realistic highlights in a surface, reduce mach band effect.
- **Disadvantages:** requires more computation, hence, more expensive rendering method.
- **Steps :**
    1. Determine the average unit normal vector at each polygon vertex.
    2. Linearly interpolate the vertex normal over the surface of the polygon.
    3. Apply an illumination model along each scan line to calculate projected pixel intensities for the surface points.



Gouraud Shading

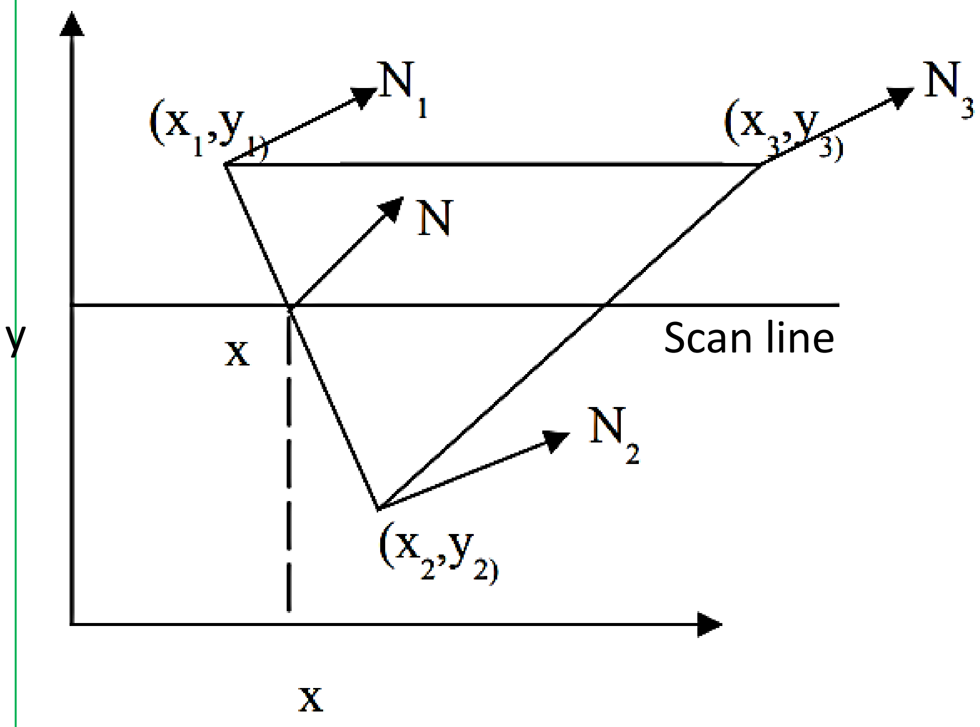


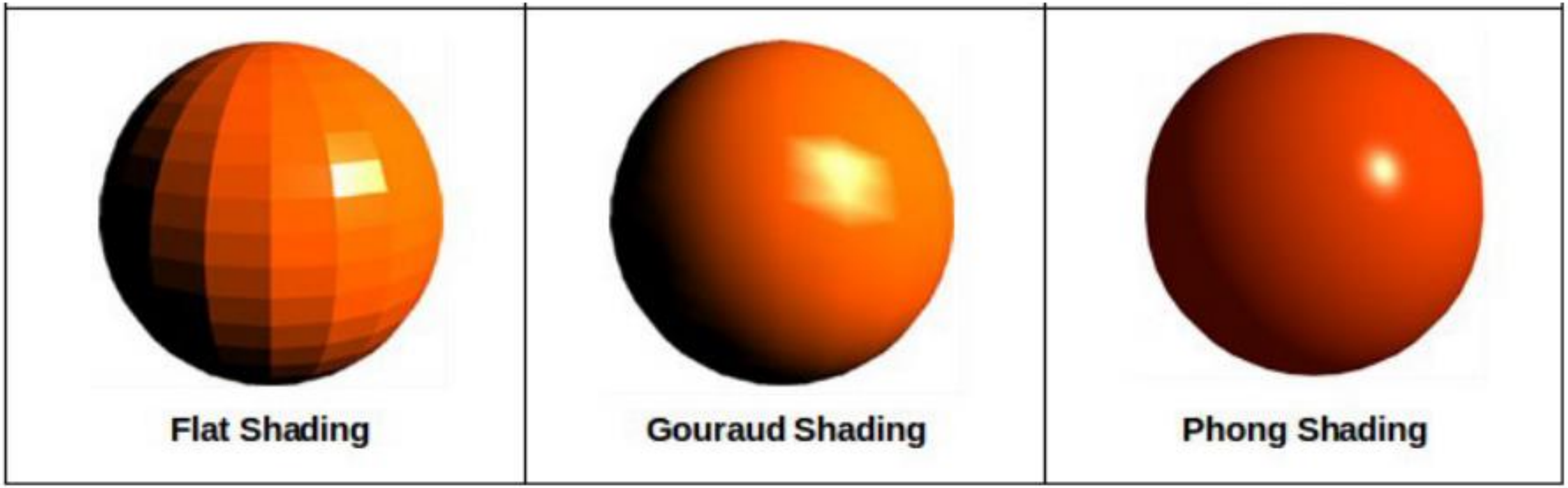
Phong Shading

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

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

- Then, incremental methods are used to evaluate normal between scanlines and along each individual scanline.
- This method however involves more calculations.




Flat Shading
Gouraud Shading
Phong Shading

d) Fast Phong Shading

- 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.

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

- Omitting the reflectivity and attenuation parameters:

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

Assignment:

- Differentiate between Gouraud shading and Phong shading

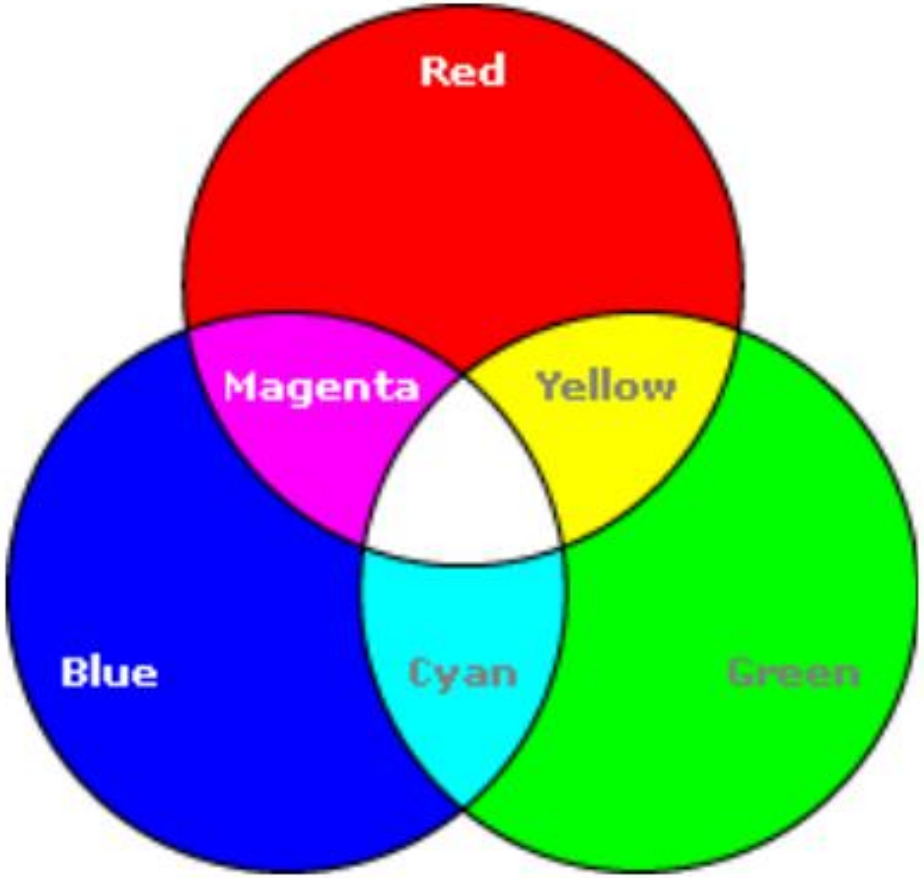
5.3 Color Models

5.3 Color Models: RGB, CMYK

- A color model is simple way to define color.
- A model describes how color will appear on the computer screen or on paper.
- Two popular color models are:
  1. RGB (Red-Green-Blue)
  2. CYMK (Cyan-Magenta-Yellow-Black)

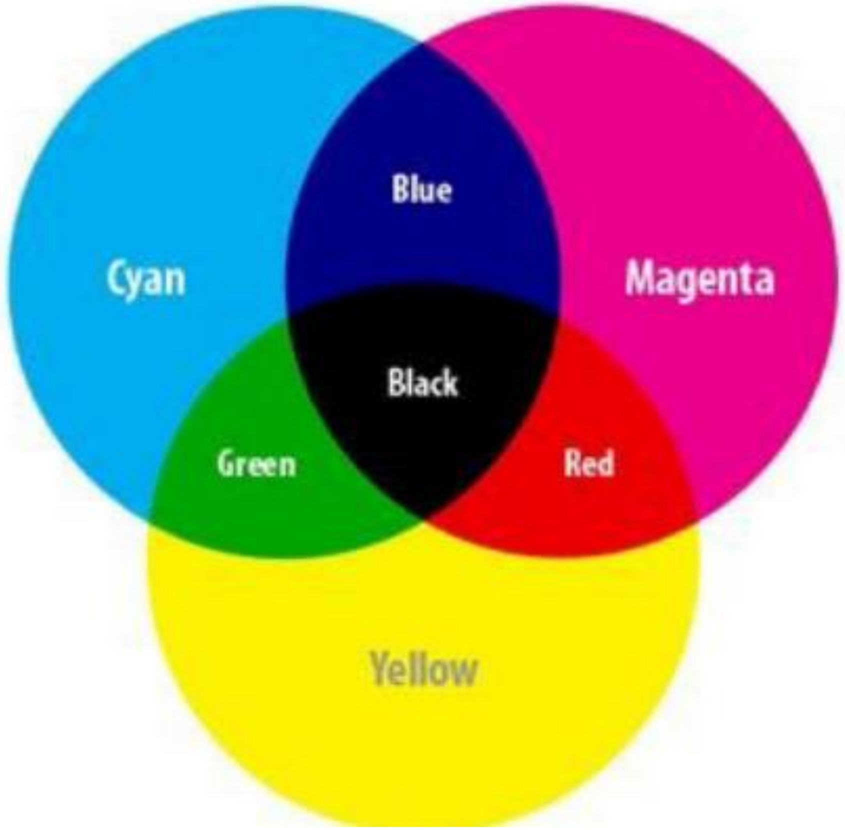
i) RGB

- The RGB model is used when working with screen based designs.
- A value between 0 and 255 is assigned to each of the light colors: Red, green and Blue.
- So for example, if we want to create a purely blue color, then Red value would have a value of 0, Green 0 and Blue would have a value of 255 (pure blue).
- To create Black: Red=0, Green=0 and Blue=0
- To create white: Red=255, Green=255 and Blue=255
- RGB is known as Additive model.
- So called because primary colors intensities are added to produce other colors.



ii) CMYK model

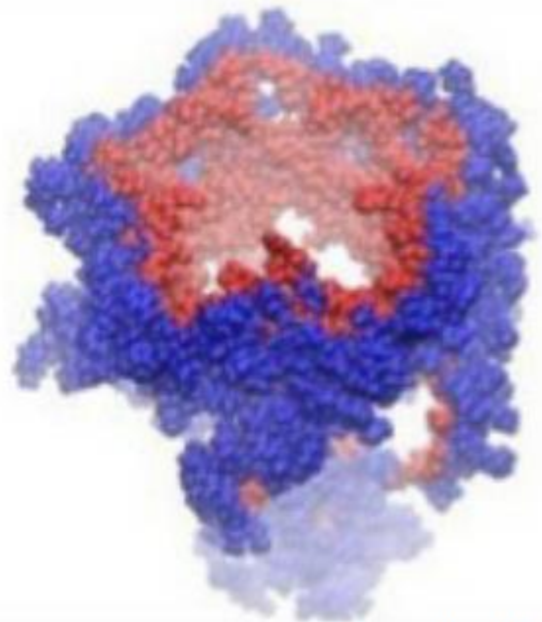
- It describes color based on the percentage of Cyan, Magenta, Yellow and Black.
- These four colors are needed to reproduce full color artwork in magazines, books and brochures.
- CMYK is used for print work. These four color are used by commercial printers in print house, press manufacturers. Etc.
- The CMYK model works by partially or entirely masking certain colors on the typically white background (i.e. absorbing particular wavelengths of light).
- Such a model is called **subtractive** because ink subtracts brightness from white.
- In CMYK, white is the natural color of the paper or other background, while black results from a full combination of all colored inks (i.e. Cyan, Magenta and Yellow).



Depth Cueing

It is the process of rendering distant objects at a lower intensity than near ones, hence giving them the appearance of depth.

Depth cuing indicates the process in which the lines closest to the viewing position are displayed with the highest intensities, and lines farther away are displayed with decreasing intensities. Depth cueing is applied by choosing maximum and minimum intensity (or color) values and a range of distances over which the intensities are to vary.



Exam Questions:

Exam Questions: Hidden Surface Removal

1. What is hidden surface problem? Write scan line algorithm for same. [2011 fall]
2. What do you mean by visible surface detection method? Describe Z-buffer method. [2011 spring]
3. Why polygon is called standard graphics object? Explain Z-buffer method for hidden surface removal technique. [2012 fall]
4. Write down the drawbacks of Backface detection. Explain z-buffer algorithm. [2012 spring]
5. Write z-buffer algorithm for detecting visible surface with its drawback and remedy. [2013 fall]
6. What do you mean by hidden surface removal? Discuss scan line method for removing hidden surfaces. [2013spring]
7. Compare object space method with image space method. Explain scan line algorithm for detecting visible surfaces with suitable figure. [2014 fall]
8. Compare object space method with image space method. Explain scan line algorithm for detecting visible surfaces with suitable figure. [2014 spring/ 2017 fall]]
9. Why is it required to take care of issues like removal of hidden surfaces in 3D viewing? Differentiate between A buffer and Depth Sorting Approach for detecting visible surfaces in 3D? [2015 fall]
10. How do the ISM approaches differ from OSM approaches for detecting visible surfaces in 3D? Differentiate between Area subdivision method and Depth sorting approach for detecting visible surfaces in 3D? [2015 spring]
11. Differentiate between Image space method and object space method? Also write down the Painter's algorithm. [2016 fall]
12. Write the z-buffer algorithm for detecting visible surface with its drawback and remedy. [2016 fall]

13. Giving the comparison of depth value, explain the depth buffer algorithm for detecting visible surface. What is its drawback? How is it removed? [2016 spring]
14. Explain back face detection with an example. [2017 spring]
15. Define the terms: depth cueing and surface rendering. Write down the necessary algorithms for any one of the image space method. [2018 fall]
16. Explain the steps in the z-buffer algorithm. [2018 spring]
17. Distinguish between Image space method and object space method. How A-buffer method removes the drawback of Z-buffer method? [2019 fall]
18. What is hidden surface? Explain the back face detection method. [2019 spring]
19. Explain the importance of hidden surface removal in computer graphics, explain scan line method of hidden surface removal. [2020 fall]
20. Explain the importance of hidden surface removal in computer graphics. What are the drawbacks of z-buffer method and how it is corrected in A-buffer? [2021 fall]
21. Why is it necessary to detect visible surfaces, in case of 3D viewing? Explain z-buffer algorithm for hidden surface removal. [2021 spring]
22. Why depth sorting is called Painter's algorithm? Explain scan line method for visible surface detection with an example. [2022 fall]
23. Write short notes:
    a) A-buffer algorithm [2014 fall]
    b) Depth buffer method [2017 fall]

Exam Questions: Illumination & shading

1. Derive an equation for calculating the total intensity due to specular reflections. [2011 fall]
2. Explain Gouraud shading method. How is it different from Phong shading method? [2012 spring]
3. What is surface rendering? Explain the Gouraud shading method for surface rendering. [2013 fall]
4. What is surface rendering? Explain Phong shading method for surface rendering. [2013 spring]
5. What is diffuse light? Derive the equations to calculate the intensity of diffuse reflection. [2014 fall]
6. Explain the Constant Gouraud and Phong shading models. [2014 spring]
7. Define lighting model and ambient light. Differentiate Phong shading and Gouraud shading method. [2015 fall]
8. How does the Gouraud shading algorithm interpolate intensities at different points of a polygon surface to give a smooth shading effect? [2015 fall]
9. At what time which color models (RGB and CMYK) is important? Explain. [2015 spring]
10. Explain the expression used for calculating the intensity of light incident on a surface due to specular reflection. How is intensity interpolated in case of Gouroud shading? [2016 fall]
11. Differentiate between RGB and CMYK color models. Explain any two graphical file formats. [2016 spring]
12. What is Gouroud shading? Explain it with an example. What are its drawbacks? [2016 spring]
13. Explain Gouraud shading and Phong shading technique in detail with their advantages and disadvantages. [2017 fall]

Exam Questions:

14. What is ambient light? Compare diffuse reflection with specular reflection. [2017 spring]
15. Explain Fast Phong shading algorithm inn detail with necessary equations and figures. [2017 spring]
16. Obtain an illumination equation due to ambient, diffused and specular reflection model at a point. [2018 fall]
17. Explain the Gouroud shading method with its advantages. [2018 spring]
18. Why is RGB called as additive and CMYK called as subtractive model? [2018 spring]
19. What do you mean by ambient light? Compare between Additive and Subtractive color model. [2019 fall]
20. Explain ambient, diffuse and specular reflection. [2019 spring]
21. What is color model. Explain RGB and CMYK color model. [2020 fall]
22. Define color model in CG. Differentiate between additive and subtractive color. [2021fall]
23. What is Mach band effect? Differentiate between Gouraud and Phong shading. [2021 fall]
24. Explain how Gouroud shading algorithm can be used in rendering a realistic 3D object. [2021 spring]
25. Write short notes on:
    a) Color model and its types [2015 fall, 2016 spring/fall, 2021 spring, 2022 fall]
    b) RGB color model [2014 fall, 2017 spring]
    c) Need of illumination model. [2020 fall]
    d) Diffuse VS Specular reflection [2021 spring]
    e) Phong shading [2022 fall]

End of Chapter